

Fundamentos de programación

php

Más de 100 algoritmos codificados

Fundamentos de programación - Estructura secuencial
Estructura selectiva simple y doble - Estructura repetitiva «Mientras»
Cadenas de caracteres

Fundamentos de programación

php

Más de 100 algoritmos codificados



España - México - Colombia - Chile - Ecuador - Perú - Bolivia - Uruguay - Guatemala - Costa Rica



Fundamentos de programación PHP

Autor: Ricardo Walter Marcelo Villalobos

© Derechos de autor registrados:

Empresa Editora Macro EIRL

© Derechos de edición, arte gráfico y diagramación reservados:

Empresa Editora Macro EIRL

Corrección de estilo:

Jorge Giraldo Sánchez

Coordinadora de edición:

Cynthia Arestegui Baca

Diseño de portada:

Alejandro Marcas León

Diagramación:

Katia Valverde Espinoza

Edición a cargo de:

© Empresa Editora Macro EIRL

Av. Paseo de la República N.° 5613, Miraflores, Lima, Perú

☎ Teléfono: (511) 748 0560

✉ E-mail: proyecto@editorialmacro.com

🌐 Página web: www.editorialmacro.com

Primera edición: octubre de 2008

Segunda edición: octubre de 2014

Tiraje: 1000 ejemplares

Impresión

Talleres gráficos de la Empresa Editora Macro EIRL

Jr. San Agustín N.° 612-624, Surquillo, Lima, Perú

ISBN N.° 978-612-304-237-0

Hecho el depósito legal en la Biblioteca Nacional del Perú N.° 2014-15126

Prohibida la reproducción parcial o total, por cualquier medio o método, de este libro sin previa autorización de la Empresa Editora Macro EIRL.

AUTOR

Ricardo Marcelo Villalobos

Profesional de sistemas y contabilidad, con más de diez años de experiencia en TI, ha participado como asesor y desarrollador en proyectos de *software* para diversas empresas privadas y públicas del país, como Minera del Hill, Aruntani, Verkaufen, MINSA, IPD; y transnacionales como Magna Rosseta Cerámica - MRC, en la cuales ha utilizado sus conocimientos de contabilidad y de ingeniería de *software* para realizar el análisis y diseños de *software* con RUP, UML y patrones de arquitectura. Asimismo, ha realizado diseños con lenguajes Java, .NET y PHP; también ha trabajado con base de datos Oracle, SQL Server, MySQL y PostgreSQL.

Asimismo, ha participado como expositor en diferentes universidades e institutos (Universidad Nacional de Ingeniería - CEPS-UNI, Universidad Nacional de Trujillo, Universidad César Vallejo de Trujillo, Universidad Nacional José Faustino Sánchez Carrión de Huacho, Instituto San Agustín, Instituto José Pardo, Instituto Manuel Seoane Corrales, Instituto La Reyna Mercedaria). Ha escrito libros, artículos y manuales de desarrollo de *software* (*Visual Basic Nivel III componentes, Oracle 10g*, manuales de VB.NET, ADO.NET, POO.NET, Access, Java POO, PHP Fundamentos, PHP POO).

En 2008 fue invitado por la Empresa Editora Macro para formar parte del *staff* de escritores, y salen a la luz cuatro obras relacionadas a los primeros pasos de la ingeniería de *software* (libros de fundamentos y más de 100 algoritmos con Visual Basic, Java, C++ y C#).

En la actualidad, difunde su experiencia como docente en la Universidad Nacional de Ingeniería (UNI-FIIS - CEPS-UNI) y el Instituto San Ignacio (ISIL); asimismo, realiza capacitaciones para empresas, como Telefónica del Perú, FAP, la Caja de Pensiones Militar Policial, ALPECO, Banco de Materiales, entre otros.

Agradecimiento

Es difícil dejar de mencionar a las personas que día a día fortalecen el conocimiento y la sabiduría de los demás. Me faltarían líneas en este libro para mencionar a todos, pero quiero agradecer, en primer lugar, a Dios y a mis padres.

También a personas muy especiales que, con su sabiduría y experiencia, han ayudado y permitido plasmar muchas de sus ideas en esta obra: Peggy Sánchez, Sergio Matsukawa, Gustavo Coronel, Gino Henostroza, Julio Flores, Joel Carrasco, Luis Zúñiga, Jesús Echevarría y todos mis alumnos y amigos en general.

PRÓLOGO

Prólogo

Cómo no recordar las primeras clases de Algoritmo y la ilusión que todos tienen por aprender a programar. Esta obra plasma los primeros pasos que cualquier estudiante de la carrera de Ingeniería de Sistemas, *Software* e Informática debe conocer para empezar a analizar, diseñar y codificar sus primeros algoritmos; y así pasar la barrera que todo programador debe dominar, que son las estructuras de control de flujo tales como *if*, *switch* (C++, Java y C#), *select case* (vb), *while* y *for*.

Este libro contiene nueve capítulos con más de cien algoritmos resueltos y otros ochenta propuestos; estoy seguro de que al concluir la lectura, el usuario formará parte del mundo de los desarrolladores de *software*. En el primer capítulo se desarrollan los conceptos generales de arquitectura de la PC, *hardware*, *software*, lenguajes de programación, metodología de algoritmos, diagramas de flujo, pseudocódigo, variables, constantes, instrucciones, entre otros.

El segundo apartado contiene diez algoritmos básicos para entender y resolver en forma simple los problemas de entrada, proceso (secuencial) y salida de los cálculos realizados. El tercer capítulo presenta quince algoritmos con las estructuras más utilizadas en la solución de problemas, llamada *if*. En el cuarto capítulo se explica la forma más fácil de solucionar problemas sin el uso de *if* anidados y engorrosos. En el quinto capítulo se enseña a entender y dominar la estructura repetitiva, y a aplicar los conceptos de contador, acumulador, bucles, entre otros.

Debido a que muchas veces es más fácil resolver procesos repetitivos usando la estructura *for*, en el sexto apartado se encuentran quince problemas resueltos; aunque muchos de ellos pertenecen al capítulo anterior, esto servirá para analizar su simplicidad. En el séptimo apartado –tomando en cuenta que uno de los temas más utilizados en el manejo de colecciones de datos tiene que ver con los arreglos (*arrays*)– se explica el concepto y se resuelven problemas de arreglos, algoritmos de búsqueda y ordenación de datos. En el capítulo octavo, se explican y resuelven problemas con cadena de caracteres (texto). Finalmente, una de las mejores recomendaciones para resolver y reutilizar procesos es el concepto de *divide y vencerás*, por ello en el capítulo nueve se enseña cómo separar un problema en varias partes reutilizables.

ÍNDICE

Índice

Capítulo 1

Fundamentos de programación 13

1.1 Introducción	13
1.2 Computadora	14
1.3 Arquitectura de una computadora	14
1.4 Unidades de medida de almacenamiento	15
1.5 Sistemas de numeración	16
1.6 Conversión binario a decimal	16
1.7 Conversión decimal a binario	16
1.8 Representación de texto en el sistema binario	17
1.9 Representación binaria de datos no numéricos ni de texto	17
1.10 Los programas (<i>software</i>)	17
1.11 Lenguajes de programación	18
1.12 Traductores del lenguaje de programación	19
1.13 Ciclo de vida de un <i>software</i>	19
1.14 Algoritmo	20
1.14.1 Características que deben cumplir los algoritmos obligatoriamente	20
1.14.2 Características aconsejables para los algoritmos	21
1.14.3 Fases en la creación de algoritmos	21
1.14.4 Herramientas de un algoritmo	21
1.14.5 Instrucciones	23
1.15 Comentarios	24
1.16 Palabras reservadas	24
1.17 Identificadores	24
1.18 Variables	25
1.19 Constantes	26
1.20 Tipo de datos simples (primitivos)	26
1.21 Tipo de datos complejos (estructurados)	28
1.22 Operadores y expresiones	29
1.23 Control de flujo	32

Capítulo 2

Estructura secuencial 33

2.1 Estructura secuencial 33

Problema n.º 1 33

Problema n.º 2 35

Problema n.º 3 38

Problema n.º 4 40

Problema n.º 5 42

Problema n.º 6 44

Problema n.º 7 46

Problema n.º 8 48

Problema n.º 9 51

Problema n.º 10 53

2.2 Problemas propuestos 56

Capítulo 3

Estructura selectiva simple y doble 57

3.1 Introducción 57

3.2 Estructura selectiva simple 57

3.3 Estructura selectiva doble 58

3.4 Estructuras anidadas 58

Problema n.º 11 59

Problema n.º 12 62

Problema n.º 13 65

Problema n.º 14 68

Problema n.º 15 71

Problema n.º 16 73

Problema n.º 17 76

Problema n.º 18 78

Problema n.º 19 82

Problema n.º 20 84

Problema n.º 21 88

Problema n.º 22 91

Problema n.º 23 95

Problema n.º 24 97

Problema n.º 25 99

3.5 Problemas propuestos 103

Capítulo 4

Estructura selectiva múltiple 105

4.1 Introducción 105

4.2 Estructura selectiva múltiple 105

 4.2.1 Estructura selectiva múltiple usando rangos 107

Problema n.º 26 108

Problema n.º 27 111

Problema n.º 28 114

Problema n.º 29 117

Problema n.º 30 120

Problema n.º 31 123

Problema n.º 32 126

Problema n.º 33 129

Problema n.º 34 132

Problema n.º 35 135

Problema n.º 36 140

Problema n.º 37 143

Problema n.º 38 146

Problema n.º 39 149

Problema n.º 40 152

4.3 Problemas propuestos 159

Capítulo 5

Estructura repetitiva «Mientras» 161

5.1 Introducción 161

5.2 Contador 161

5.3 Acumulador 162

5.4 Salir del bucle 162

5.5 Continuar al inicio del bucle 162

5.6 Estructura repetitiva «Mientras» 163

5.7 Estructura repetitiva «Mientras» anidada 163

Problema n.º 41 164

Problema n.º 42 166

Problema n.º 43 168

Problema n.º 44 170

Problema n.º 45 172

Problema n.º 46 174

Problema n.º 47 176

Problema n.º 48 178

Problema n.º 49	181
Problema n.º 50	183
Problema n.º 51	185
Problema n.º 52	188
Problema n.º 53	190
Problema n.º 54	193
Problema n.º 55	196
5.8 Problemas propuestos	199

Capítulo 6

Estructura repetitiva «Para» 201

6.1 Introducción	201
6.2 Estructura repetitiva «Para»	201
6.3 Estructura repetitiva «Para» anidada	202
Problema n.º 56	202
Problema n.º 57	204
Problema n.º 58	206
Problema n.º 59	209
Problema n.º 60	210
Problema n.º 61	213
Problema n.º 62	214
Problema n.º 63	217
Problema n.º 64	220
Problema n.º 65	222
Problema n.º 66	224
Problema n.º 67	227
Problema n.º 68	229
Problema n.º 69	231
Problema n.º 70	234
6.4 Problemas propuestos	237

Capítulo 7

Estructuras de datos. Arreglos (vectores y matrices) 239

7.1 Introducción	239
7.2 <i>Arrays</i> (arreglos)	240
7.3 Operaciones con <i>arrays</i>	240
7.4 Creación de <i>arrays</i>	241
7.5 Recorrido por los elementos del <i>array</i>	242
Problema n.º 71	243

Problema n.º 72 245

Problema n.º 73 248

Problema n.º 74 251

Problema n.º 75 254

Problema n.º 76 257

Problema n.º 77 260

Problema n.º 78 263

Problema n.º 79 265

Problema n.º 80 268

Problema n.º 81 272

Problema n.º 82 276

Problema n.º 83 279

Problema n.º 84 282

Problema n.º 85 285

7.6 Problemas propuestos 292

Capítulo 8

Cadenas de caracteres 293

8.1 Introducción 293

8.2 Juego de caracteres 293

8.3 Carácter (*char*) 294

8.4 Cadena de caracteres (*string*) 295

8.5 Operaciones con cadena 295

8.6 Concatenación 295

8.7 Comparación 296

8.8 Cálculo de longitud 296

8.9 Extracción de cadenas (subcadenas) 297

8.10 Búsqueda de cadenas 298

8.11 Conversiones 298

Problema n.º 86 299

Problema n.º 87 301

Problema n.º 88 303

Problema n.º 89 304

Problema n.º 90 307

Problema n.º 91 309

Problema n.º 92 311

Problema n.º 93 313

Problema n.º 94 315

Problema n.º 95 318

8.12 Problemas propuestos 321

Capítulo 9

SubAlgoritmos (procedimientos y funciones) 323

9.1 Introducción 323

9.2 Procedimientos 324

9.3 Funciones 324

9.4 Paso de parámetros 325

9.5 Parámetros por valor (entrada) 325

9.6 Parámetros por referencia (salida) 326

Problema n.º 96 327

Problema n.º 97 330

Problema n.º 98 333

Problema n.º 99 336

Problema n.º 100 339

Problema n.º 101 342

Problema n.º 102 346

Problema n.º 103 350

Problema n.º 104 354

9.7 Problemas propuestos 358

Fundamentos de programación

1.1 Introducción

En los primeros ciclos de toda carrera profesional relacionada a la ingeniería de sistemas, los estudiantes requieren entender, aprender y dominar los fundamentos de programación para resolver problemas que permitirán automatizar procesos usando la computadora.

Saber programar es la base de toda su carrera y, para conseguir este objetivo, he plasmado mi experiencia de docencia de mas de diez años en el campo de la Ingeniería de Sistemas. Sé que este libro le ayudará a resolver todas sus dudas y dominar las principales estructuras de programación.

Este libro contiene más de 100 algoritmos resueltos y codificados en el lenguaje C#, que en la actualidad es uno de los lenguajes de programación propuesto por Microsoft para la tecnología .NET



A continuación se describen los conceptos generales de los fundamentos de programación.

1.2 Computadora

Es un aparato electrónico que recibe datos (entrada), los procesa (instrucciones denominado programa) y devuelve información (salida), también conocido como ordenador o PC (Personal Computer). En la actualidad existe una variedad de computadoras para diferentes propósitos.

Servidores



Computadora personal



Computadora Portátil



PDA



1.3 Arquitectura de una computadora

Las computadoras tienen dos componentes principales que son el **hardware** y el **software**, que trabajan en coordinación para llevar a cabo sus objetivos.

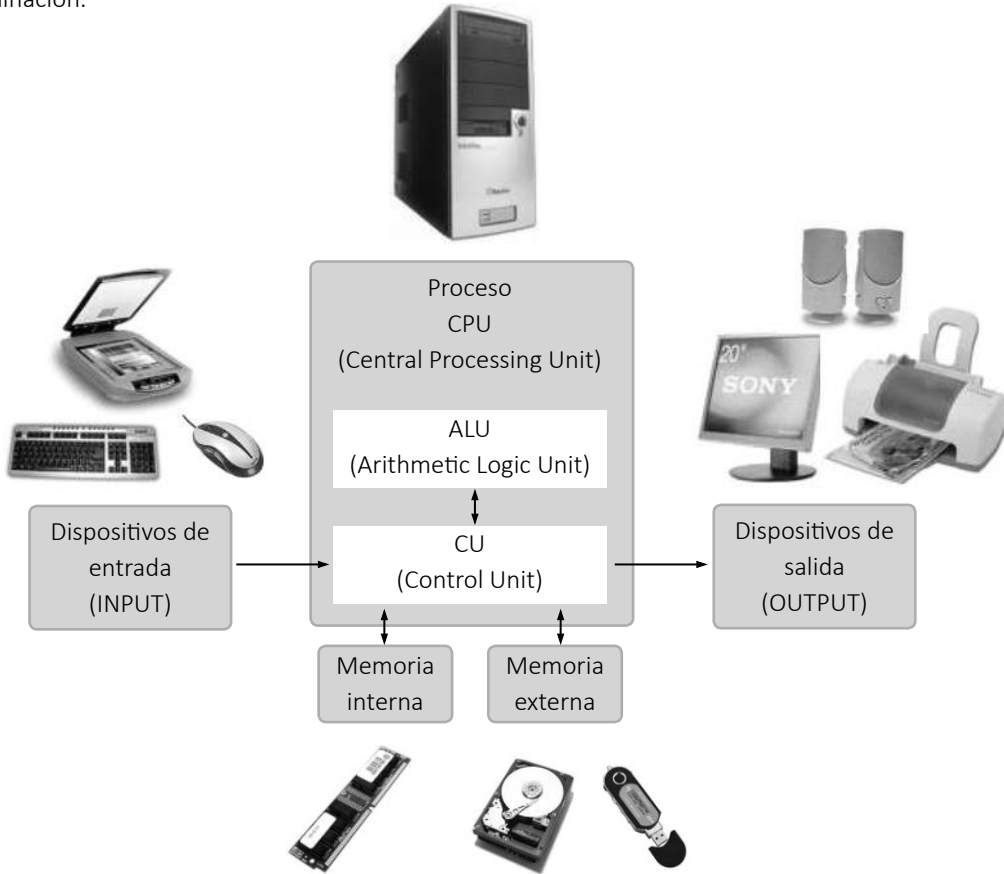
Hardware: *Hard* (duro) – *ware* (componente), representa la parte física de la computadora.



Software: *Soft* (blando) – *ware* (componente), representa la parte lógica de la computadora (los programas); estos se encuentran almacenados en los componentes físicos de la computadora, tales como memorias RAM, ROM, Discos Duros (*Hard Disk*), entre otros.



La siguiente figura muestra la arquitectura de la computadora y sus principales componentes en coordinación.



1.4 Unidades de medida de almacenamiento

La memoria interna (RAM) y las memorias externas (disco duro) almacenan información. La información que se guarda y entiende la PC está en formato binario (0- 1).

BIT (Binary Digit): El bit representa la unidad mínima de información que almacena una computadora.

BYTE: Está compuesto por 8 bit (01110011), entonces existe $2^8 = 256$ combinaciones diferentes (tabla de código ASCII).

Por lo general, la información se representa por caracteres y cada carácter (número, letra, símbolo, etc.) es un byte. Para medir la información se utilizan múltiplos de bytes.

Byte	1 B		8 bits
Kilobyte	1 KB	2^{10} bytes	1024 bytes
Megabyte	1 MB	2^{20} bytes	1024 KB
Gigabyte	1 GB	2^{30} bytes	1024 MB
Terabyte	1 TB	2^{40} bytes	1024 GB

1.5 Sistemas de numeración

Todos los sistemas de numeración tienen una **base**, que es el número total de símbolos que utiliza el sistema. En el caso de la numeración decimal, la base es 10; en el sistema binario es 2.

El **Teorema Fundamental de la Numeración** permite saber el valor decimal que tiene cualquier número en cualquier base. Dicho teorema utiliza la fórmula:

$$\dots + X_3 \cdot B^3 + X_2 \cdot B^2 + X_1 \cdot B^1 + X_0 \cdot B^0 + X_{-1} \cdot B^{-1} + X_{-2} \cdot B^{-2} + \dots$$

Donde:

- **X_i :** Es el símbolo que se encuentra en la posición número i del número que se está convirtiendo. Teniendo en cuenta que la posición de las unidades es la posición 0 (la posición -1 sería la del primer decimal).
- **B :** Es la base del sistemas que se utiliza para representar al número.

Por ejemplo si tenemos el número 153,6 utilizando el sistema octal (base ocho), el paso a decimal se haría:

$$1 \cdot 8^2 + 5 \cdot 8^1 + 3 \cdot 8^0 + 6 \cdot 8^{-1} = 64 + 40 + 3 + 6 \div 8 = 107,75$$

1.6 Conversión binario a decimal

El **Teorema Fundamental de la Numeración** se puede aplicar para saber el número decimal representado por un número escrito en binario. Así, para el número binario 10011011011 la conversión sería (los ceros se han ignorado):

$$1 \cdot 2^{10} + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1243$$

1.7 Conversión decimal a binario

El método más utilizado consiste en ir haciendo divisiones sucesivas entre dos. Los restos son las cifras binarias. Por ejemplo, para pasar el 39:

$$\begin{array}{l} 39 \div 2 = 19 \text{ resto } \mathbf{1} \\ 19 \div 2 = 9 \text{ resto } \mathbf{1} \\ 9 \div 2 = 4 \text{ resto } \mathbf{1} \\ 4 \div 2 = 2 \text{ resto } \mathbf{0} \\ 2 \div 2 = 1 \text{ resto } \mathbf{0} \\ 1 \div 2 = 0 \text{ resto } \mathbf{1} \end{array}$$

Ahora las cifras binarias se toman al revés. Con lo cual, el número 100111 es el equivalente en binario de 39.

1.8 Representación de texto en el sistema binario

Puesto que una computadora no solo maneja números, habrán dígitos binarios que contengan información no traducible al sistema decimal. Todo depende de cómo se interprete esa traducción. Por ejemplo, en el caso del texto, lo que se hace es codificar cada carácter en una serie de números binarios. El código **ASCII** ha sido durante mucho tiempo el más utilizado, inicialmente era un código que utilizaba 7 bits para representar texto, lo que significaba que era capaz de codificar 127 caracteres. Por ejemplo, el número 65 (1000001 en binario) se utiliza para la **A** mayúscula.

Poco después apareció un problema: este código bastaba para los caracteres del inglés, pero no para otras lenguas. Entonces se añadió el octavo bit para representar otros 128 caracteres que son distintos, según idiomas (Europa Occidental usa unos códigos que no utiliza Europa Oriental).

Eso provoca que un código como el 190 signifique cosas diferentes si cambiamos de país. Por ello, cuando un ordenador necesita mostrar texto, tiene que saber qué juego de códigos debe de utilizar, lo cual supone un tremendo problema.

Una ampliación de este método de codificación es el código **UNICODE**, que puede utilizar hasta 4 bytes (32 bits), con lo que es capaz de codificar cualquier carácter en cualquier lengua del planeta, utilizando el mismo conjunto de códigos. Poco a poco se ha ido extendiendo cada vez más, pero la preponderancia histórica que ha tenido el código ASCII complica su popularidad.

1.9 Representación binaria de datos no numéricos ni de texto

En el caso de datos más complejos (imágenes, vídeo, audio) se necesita una codificación más compleja. Además, en estos datos no hay estándares, por lo que hay decenas de formas de codificar. En el caso, por ejemplo, de las imágenes, una forma básica de codificarlas en binario es la que graba cada **píxel** (cada punto distinguible en la imagen) mediante **tres bytes**: el primero graba el nivel de **rojo**; el segundo, el nivel de **azul**; y el tercero, el nivel de **verde**. Y así por cada píxel.

Por ejemplo, un punto en una imagen de color rojo puro:

11111111 00000000 00000000

Naturalmente, en una imagen no solo se graban los píxeles sino el tamaño de la imagen, el modelo de color, etc. De ahí que representar estos datos sea tan complejo para el ordenador (y tan complejo entenderlo para nosotros).

1.10 Los programas (**software**)

Los programas o **software** son un conjunto de instrucciones ordenadas para ejecutarse de forma rápida y precisa en una computadora. El **software** se divide en dos grupos: **software** de **sistema operativo** y **software de aplicaciones**.

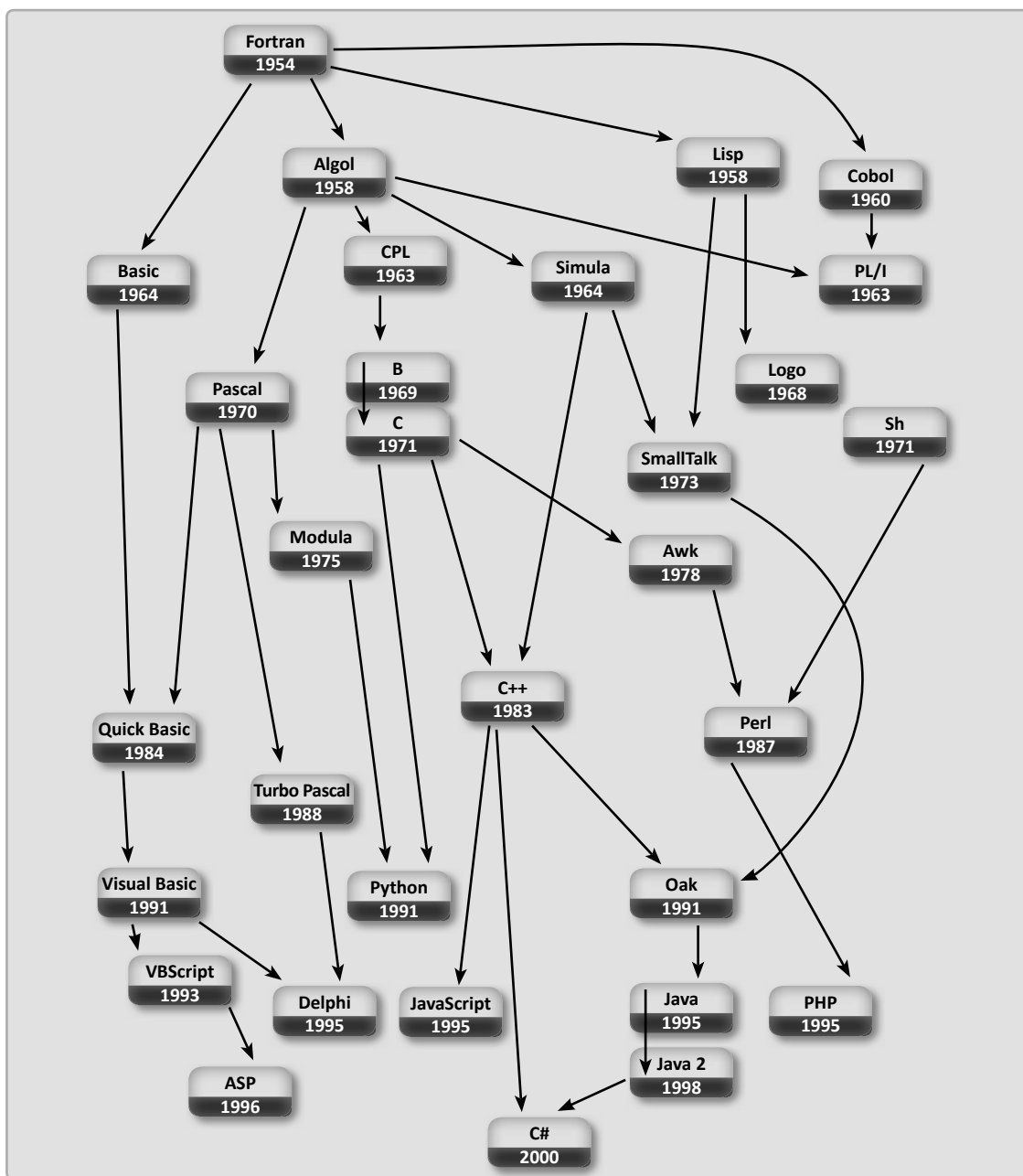
El proceso de escribir un programa se denomina **programación**, y el conjunto de instrucciones que se utilizan para escribir un programa se llama **lenguaje de programación**.

1.11 Lenguajes de programación

Sirve para escribir programas y permite la comunicación usuario (programador) versus máquina (PC).

Existen tres tipos de lenguajes de programación:

- **Lenguaje de máquina:** Programación binaria, difícil de programar y dependiente de la máquina.
- **Lenguaje de bajo nivel (ensamblador):** Usa símbolos nemotécnicos, necesita ser traducido al lenguaje de máquina y sigue siendo dependiente.
- **Lenguaje de alto nivel:** Cercano al lenguaje natural, tiempo de programación relativamente corto, es independiente de la máquina. A continuación se muestra un plano de la evolución de los lenguajes de programación de alto nivel.



1.14 Algoritmo

Método que describe la solución de un problema computacional mediante una serie de pasos precisos, definidos y finitos.

- **Preciso:** Indicar el orden de realización en cada paso.
- **Definido:** Al repetir los pasos n veces se obtiene el mismo resultado.
- **Finito:** Tiene un número determinado de pasos.

La solución de un algoritmo debe describir tres partes:

- **Entrada:** Datos que se necesitan para poder ejecutarse.
- **Proceso:** Acciones y cálculos a realizar.
- **Salida:** Resultado esperado.



La palabra algoritmo procede del matemático árabe **Mohamed Ibn Al Kow Rizmi**, quien escribió entre los años 800 y 825 su obra *Quitad Al Mugabala*, donde recogió el sistema de numeración hindú y el concepto del cero. **Fibonacci**, tradujo la obra al latín y la llamó *Algoritmi Dicit*.

El lenguaje algorítmico es aquel que implementa una solución teórica a un problema, indicando las operaciones a realizar y el orden en el que deben efectuarse. Por ejemplo, en el caso de que nos encontremos en casa con un foco malogrado de una lámpara, un posible algoritmo sería:

- a. Comprobar si hay foco de repuesto.
- b. En el caso de que haya, sustituir el foco anterior por el nuevo.
- c. Si no hay foco de repuesto, bajar a comprar uno nuevo en la tienda y ponerlo en lugar del malogrado.

Los algoritmos son la base de la programación de ordenadores, ya que los **programas de ordenador** se pueden entender como algoritmos escritos en un código especial, entendible por un ordenador.

La desventaja del diseño de algoritmos radica en que no podemos escribir lo que deseamos; el lenguaje ha utilizar no debe dejar posibilidad de duda, debe recoger todas las posibilidades.

1.14.1 Características que deben cumplir los algoritmos obligatoriamente

- **Un algoritmo debe resolver el problema para el que fue formulado.** Lógicamente, no sirve un algoritmo que no resuelve ese problema. En el caso de los programadores, a veces crean algoritmos que resuelven problemas diferentes al planteado.
- **Los algoritmos son independientes del lenguaje de programación.** Los algoritmos se escriben para poder ser utilizados en cualquier lenguaje de programación.
- **Los algoritmos deben ser precisos.** Los resultados de los cálculos deben ser exactos, de manera rigurosa. No es válido un algoritmo que sólo aproxime la solución.
- **Los algoritmos deben ser finitos.** Deben de finalizar en algún momento. No es un algoritmo válido aquel que produce situaciones en las que el algoritmo no termina.
- **Los algoritmos deben poder repetirse.** Deben permitir su ejecución las veces que haga falta. No son válidos los que tras ejecutarse una vez ya no pueden volver a hacerlo por la razón que sea.

1.14.2 Características aconsejables para los algoritmos

- **Validez:** Un algoritmo es válido si carece de errores. Un algoritmo puede resolver el problema para el que se planteó y, sin embargo, no ser válido debido a que posee errores.
- **Eficiencia:** Un algoritmo es eficiente si obtiene la solución al problema en poco tiempo. No lo es si tarda en obtener el resultado.
- **Óptimo:** Un algoritmo es óptimo si es el más eficiente posible y no contiene errores. La búsqueda de este algoritmo es el objetivo prioritario del programador. No siempre podemos garantizar que el algoritmo hallado sea el óptimo, a veces sí.

1.14.3 Fases en la creación de algoritmos

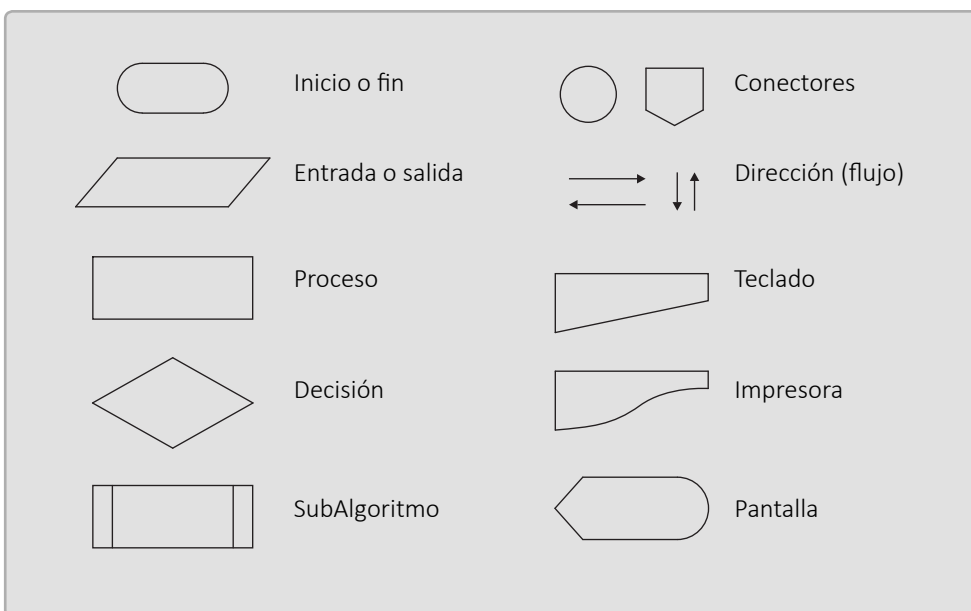
Hay tres fases en la elaboración de un algoritmo:

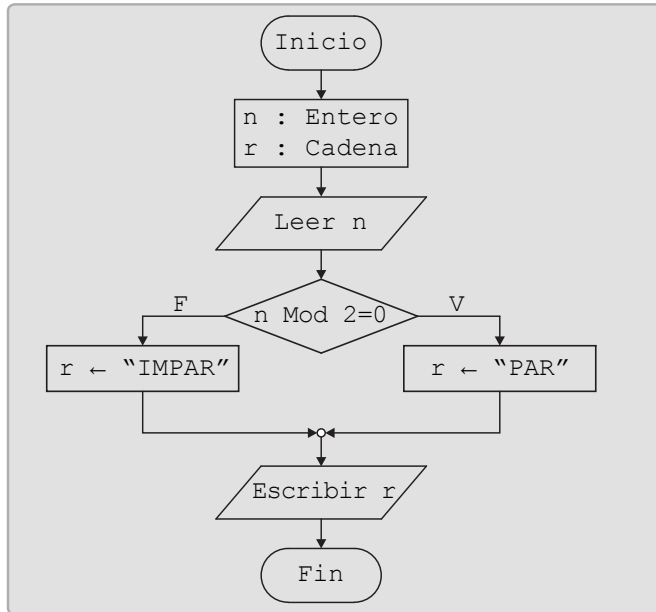
- Análisis:** En esta se determina cuál es exactamente el problema a resolver. Qué datos forman la entrada del algoritmo y cuáles deberán obtenerse como salida.
- Diseño:** Elaboración del algoritmo.
- Prueba:** Comprobación del resultado. Se observa si el algoritmo obtiene la salida esperada para todas las entradas.

1.14.4 Herramientas de un algoritmo

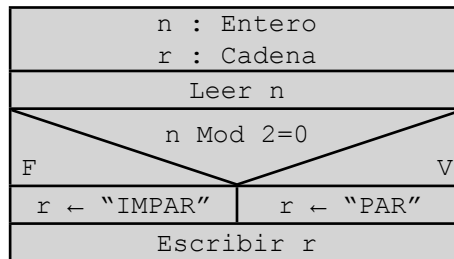
Para expresar la solución de un problema se pueden usar diferentes herramientas de programación, tales como diagrama de flujo (*flow chart*), diagrama N-S (Nassi Schneiderman), pseudocódigo.

- **Diagrama de flujo:** Es una representación gráfica que utiliza símbolos normalizados por ANSI, y expresa las sucesivas instrucciones que se deben seguir para resolver el problema. Estas instrucciones no dependen de la sintaxis de ningún lenguaje de programación, sino que deben servir fácilmente para su transformación (codificación) en un lenguaje de programación.





- **Diagrama de Nassi Scheneiderman (N-S):** Conocido también como el diagrama de Chapin, es como un diagrama de flujo pero sin flechas y con cajas continuas.



- **Pseudocódigo:** Permite expresar las instrucciones en un lenguaje común (ingles, español, etc.) para facilitar tanto la escritura como la lectura de la solución de un programa. No existen reglas para escribir pseudocódigo.

Inicio

//Variables

n : Entero
r : Cadena

//Entrada

Leer n

//Proceso

Si n Mod 2 = 0 Entonces
 r ← "PAR"
SiNo
 r ← "IMPAR"
Fin Si

//Salida

Escribir r

Fin

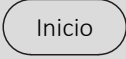
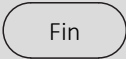
1.14.5 Instrucciones

Son las acciones que debe realizar un algoritmo para resolver un problema.

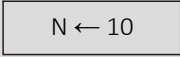
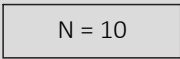
Las instrucciones más comunes son las siguientes:

- Instrucción de inicio / fin
- Instrucción de asignación.
- Instrucción de lectura.
- Instrucción de escritura.
- Instrucción de bifurcación.

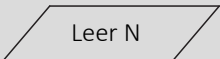
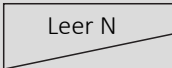
A. Instrucción de inicio/ fin: Representa el inicio y fin de un algoritmo.

Diagrama de flujo	Pseudocódigo
	Inicio
	Fin

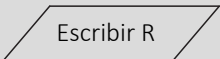
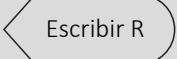
B. Instrucción de asignación: Representa la asignación de un valor a una variable, se puede representar usando una flecha o el símbolo de igualdad, el cual es usado por muchos de los lenguajes de programación.

Diagrama de flujo	Pseudocódigo
	N ← 10
	N = 10

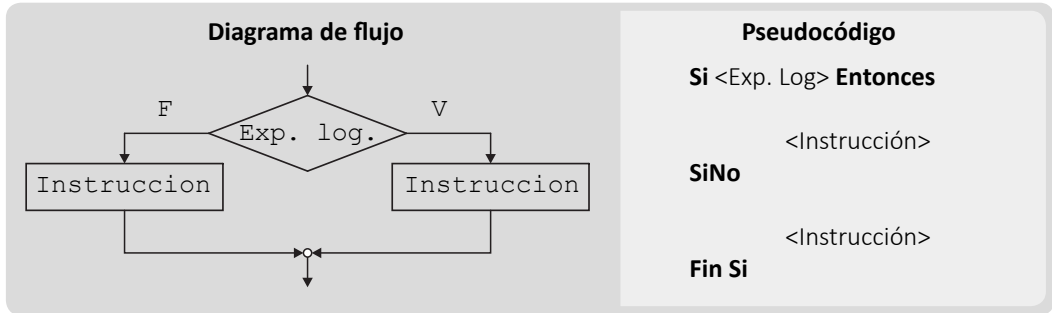
C. Instrucción de lectura: Representa el ingreso de datos mediante un dispositivo de entrada, que muchas veces es representado por un símbolo de teclado.

Diagrama de flujo	Pseudocódigo
	Leer N
	Leer N

D. Instrucción de escritura: Representa la salida de la información mediante un dispositivo de salida, puede ser representado por el símbolo de entrada/salida, por símbolo de pantalla o impresora.

Diagrama de flujo	Pseudocódigo
	Escribir R
	Escribir R
	Escribir R

E. Instrucción de bifurcación: Cambian el flujo del programa según el resultado de una expresión lógica (condición).



1.15 Comentarios

Permiten describir y explicar, además sirve como ayuda para recordar y entender las operaciones que se van a ejecutar. Los comentarios no son instrucciones, por lo tanto al ser traducido el código fuente a código binario (tiempo de compilación), los lenguajes de programación los ignoran. Dependiendo el lenguaje de programación los comentarios se escriben usando cierta simbología, en este libro usaremos el símbolo `//` en los pseudocódigos para colocar comentarios.

Ejemplo pseudocódigo

```
//Variables
N : Entero
```

PHP

```
//Variables
$N = 0
```

1.16 Palabras reservadas

Son palabras usadas por el lenguaje de programación que no deben ser utilizadas como identificadores de variables, funciones, entre otros.

Algunas de las palabras reservadas de PHP

```
int, float, double, if, for, switch, while, ...
```

1.17 Identificadores

Son los nombres que asignamos a las variables, constantes, funciones, objetos, entre otros; y no pueden coincidir con las palabras reservadas porque ocasionaría ambigüedad, y el compilador no lo entendería. Por lo general, los identificadores deben de cumplir las siguientes reglas:

- Deben comenzar por una letra. Evite usar ñ y tilde.
- No debe coincidir con palabras reservadas del lenguaje de programación que está utilizando.

Error de compilación PHP

```
// Identificador de Variable es if
// y esta es palabra reservada
$if = 0;
```

1.18 Variables

Representa un espacio de memoria RAM que guarda un valor que servirá para algún proceso en particular; dicho valor puede ser modificado en cualquier momento.

Las variables tienen, por lo general, un identificador (nombre) y, asignado, el tipo de dato que se está utilizando; es decir, si almacena un número (entero), si es texto o alfanumérico (cadena), si es un valor verdadero o falso (lógico) llamado también booleano.

PHP es un lenguaje no tipado (las variables aceptan cualquier tipo de dato) y no requiere crear previamente las variables antes de usarlas; esto permite muchas veces cometer errores si no se lleva una buena práctica de programación.

Ejemplo pseudocódigo

```
//Variables
N : Entero
```

PHP

```
//Variables
$N = 0;
```

Para asignarle un valor, usamos el operador de asignación, para algoritmos usaremos ($\leftarrow$) o ($=$); este último es el más usado por los lenguajes de programación.

Ejemplo pseudocódigo

```
//Asignar un valor
N  $\leftarrow$  10
//Cambiar su valor
N  $\leftarrow$  50
```

PHP

```
//Asignar un valor
$N = 10;
//Cambiar su valor
$N = 50;
```

1.19 Constantes

Representa un espacio de memoria RAM, el cual guarda un valor que servirá para algún proceso en particular; dicho valor permanece fijo, es decir, no puede cambiarse en la ejecución del programa. Las constantes tienen, al igual que las variables, un identificador (nombre) y un tipo de dato.

Ejemplo pseudocódigo

```
//Constantes
PI ← 3.14159 : Real
//Error ya no puede modificarlo
PI ← 3.14
```

PHP

```
//Constantes
define (PI, 3.14159);
//Error ya no puede modificarlo
PI = 3.14;
```

1.20 Tipo de datos simples (primitivos)

Al declarar una variable, debemos indicar el tipo de dato que es permitido almacenar en dicha variable. Cada lenguaje de programación trabaja con una variedad de tipo de datos; por lo general, todos usan los llamados «primitivos», que son los siguientes:

A. Entero: Representan los números enteros (no almacena decimales).

Ejemplo pseudocódigo

```
//Crear la variable
//(identificador y tipo de dato)
N : Entero
//Asignar un valor
//(identificador, operador de asignación y valor)
N ← 15
```

En el lenguaje de PHP el tipo entero es int y acepta números de un rango pequeño y grande.

Ejemplo PHP

```
//Entero
$N = 0;
//Asignar un valor
$N = 4500099;
```

B. Real: Representan los números reales (almacena decimales).

Ejemplo pseudocódigo

```
//Crear la variable
//(identificador y tipo de dato)
N : Real
//Asignar un valor
//(identificador, operador de asignación y valor)
N ← 15.75
```

En el lenguaje de PHP el tipo real es **float**, y trabaja por lo general con un máximo de 14 decimales; esto depende de la plataforma donde se está ejecutando el programa S.O. (Linux, Windows).

```
//Precisión simple
$N = 0,0;
//Decimales
$N = 15.12345678;
```

C. Carácter: Representa un carácter de cualquier tipo: texto, número, símbolo, etc. El valor se coloca entre comillas simples.

Ejemplo pseudocódigo

```
//Crear la variable
R : Carácter
//Asignar un valor
R ← 'A'
R ← '9'
R ← '*'
```

Ejemplo PHP

```
`Crear la variable  
$R = ' ' ;  
`Asignar un valor  
$R = 'A' ;  
$R = '9' ;  
$R = '*' ;
```

E. Lógico: Representan los valores «verdadero» o «falso», conocidos también como boolean, no se colocan comillas simple ni dobles.

En PHP se puede asignar el valor en mayúscula o minúscula.

Ejemplo pseudocódigo

```
//Crear la variable  
L : Logico  
//Asignar un valor  
L ← VERDADERO  
L ← FALSO
```

Ejemplo PHP

```
`Crear la variable  
$L = false;  
//Asignar un valor  
$L = True;  
$L = FALSE;
```

1.21 Tipo de datos complejos (estructurados)

Son aquellos que están constituidos por tipos de datos simples y definen una estructura de datos, un ejemplo claro es el tipo cadena, que está compuesto por un conjunto de caracteres (tipo de dato carácter). Existe una variedad de tipo de datos complejos, el enfoque de este libro es algorítmico y solo tocaremos dos tipos de datos complejos: cadena y arreglos. Los libros que profundizan el tema se llaman libros de estructura de datos.

A. Cadena: Representa un conjunto de caracteres, internamente es un arreglo de caracteres; por lo general, se representa con comillas dobles.

Ejemplo pseudocódigo

```
//Crear la variable  
R : Cadena  
//Asignar un valor  
R ← "ricardomarcelo@hotmail.com"
```

Ejemplo PHP

```
//Cadena
$R = "ricardomarcelo@hotmail.com";
```

1.22 Operadores y expresiones

Son los que permiten realizar los cálculos entre valores fijos y variables.

Los operadores se clasifican por:

- Operadores Aritméticos
- Operadores Relacionales
- Operadores Lógicos
- Operadores de Cadena

Operadores aritméticos: Son aquellos operadores que permiten realizar las operaciones aritméticas, de la misma forma como se utilizan en las matemáticas.

Operador	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División
\	División entera
^	Exponenciación
Mod	Módulo (resto de una división)

Dependiendo el lenguaje de programación los operadores varían o no implementan uno u otro, en el caso de PHP implementa los siguientes:

Operador	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo (resto de una división)

Para elevar a una potencia se usa pow (9 . 0 , 2 . 0) , dentro de los parámetros se coloca números reales (double) y para división entera use cast (conversión a entero).

Expresiones aritméticas

División real

\$N = 9 / 4; //retorna 2.25

División entera

\$N = (int)(9 / 4); //retorna 2

8×3	Equivale a	$8 * 3 = 24$
$8 \div 3$ o $\frac{8}{3}$	Equivale a	$8 / 3 = 2.666666$ $8 \backslash 3 = 2$
8^2	Equivale a	$8 ^2 = 64$
$\sqrt{9}$	Equivale a	$9 ^{(1/2)} = 3$
$\begin{array}{r} 9 \overline{) 4} \\ \textcircled{1} \end{array}$	Equivale a	$9 \text{ Mod } 4 = 1$

B. Operadores relacionales: Llamados también operadores de comparación, y permiten evaluar si dos valores guardan alguna relación entre sí.

Operador	Descripción
=	Igualdad
>	Mayor que
>=	Menor o igual que
<	Menor que
<=	Menor o Igual que
<>	Diferente a

Dependiendo del lenguaje de programación, los operadores varían o no, implementan uno u otro operador; en el caso de PHP varía la simbología en algunos.

Operador	Descripción
==	Igualdad
>	Mayor que
>=	Menor o igual que
<	Menor que
<=	Menor o Igual que
!=	Diferente a

Expresiones lógicas (condiciones)

$8 = 3$	Falso
$8 > 3$	Verdadero
$8 <= 3$	Verdadero
$8 <> 8$	Falso

C. Operadores lógicos: Son aquellos operadores que se utilizan en combinación con los operadores de relación.

Operador	Descripción
Y	Y Lógico
O	O Lógico
No	No Lógico

«Y» lógico: Si p y q son valores lógicos, ambos deben ser verdaderos para que Y devuelva verdadero.

Expresiones lógicas (condiciones)

8 > 4	Y	3 = 6	Falso
7 <> 5	Y	5 >= 4	Verdadero

«O» lógico: Si p y q son valores lógicos, uno de ellos debe ser verdadero para que O devuelva verdadero.

Expresiones lógicas (condiciones)

8 > 4	O	3 = 6	Verdadero
7 <> 5	Y	5 >= 4	Verdadero

«No» lógico: Si p es un valor lógico, el operador No invierte su valor.

Expresiones lógicas (condiciones)

NO (8 > 4)	Falso
NO (7 <> 7)	Verdadero

Para PHP se utiliza ambas simbologías.

Operador	Descripción
&& AND	Y Lógico
OR	O Lógico
! NOT	No Lógico

D. Operadores de cadena: Son aquellos operadores que permiten realizar operaciones con cadenas; por lo general, permiten unir en cadena, lo cual es llamado también «concatenar».

Operador	Descripción
+	Unir cadenas
&	Unir cadenas

Expresiones de cadena

"Ricardo" + " " + "Marcelo"	Ricardo Marcelo
"ricardomarcelo" & "@" & "hotmail.com"	ricardomarcelo@hotmail.com

En PHP se utiliza solo el símbolo (+) para unir cadenas (concatenar).

1.23 Control de flujo

Todos los lenguajes de programación implementan estructuras para controlar la ejecución de un programa, estas son:

- Estructura secuencial
- Estructura selectiva simple y doble
- Estructura selectiva múltiple
- Estructura repetitiva mientras
- Estructura repetitiva para

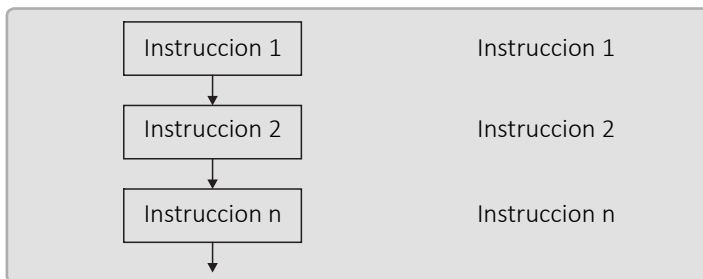
En los siguientes capítulos se explicarán cada una de las estructuras mencionadas.

Capítulo 2

Estructura secuencial

2.1 Estructura secuencial

Son aquellos algoritmos que ejecutan instrucciones en forma consecutiva, es decir, uno detrás de otro, hasta finalizar el proceso.



Problema n.º 1

Enunciado: Dado dos números enteros, hallar la suma.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese dos números enteros; luego, que el sistema realice el cálculo respectivo para hallar la suma. Para ello usará la siguiente expresión.

Expresión matemática

$$s = n1 + n2$$

Expresión algorítmica

$$s \leftarrow n1 + n2$$

Entrada

- Dos números ($n1$ y $n2$)

Salida

- La suma (s)

Diseño:

Interfaz de usuario

Problema 01 - Windows Internet Explorer

http://localhost/Cap02/Problema01.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 01

Problema 01

Numero 1 568

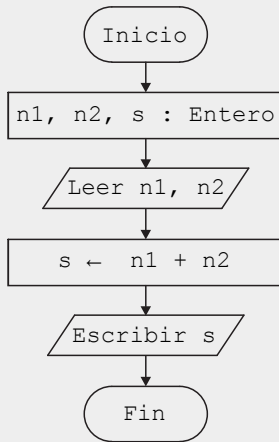
Numero 2 454

Suma 1022

Calcular

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

n1, n2, s : Entero

//Entrada

Leer n1, n2

//Proceso

s ← n1 + n2

//Salida

Escribir s

Fin

Codificación:

```

<?php

//Variables
$n1 = 0; $n2 = 0; $s = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n1 = (int)$_POST["txtn1"];
    $n2 = (int)$_POST["txtn2"];

    //Proceso
    $s = $n1 + $n2;
}
?>

<html>
<head>
<title>Problema 01</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema01.php">
    <table width="241" border="0">

```

```

<tr>
  <td colspan="2"><strong>Problema 01</strong> </td>
</tr>
<tr>
  <td width="81">Numero 1 </td>
  <td width="150">
    <input name="txtn1" type="text" id="txtn1" value="<?=$n1?>" />
  </td>
</tr>
<tr>
  <td>Numero 2 </td>
  <td>
    <input name="txtn2" type="text" id="txtn2" value="<?=$n2?>" />
  </td>
</tr>

<tr>
  <td>Suma</td>
  <td>
    <input name="txts" type="text" class="TextoFondo" id="txts"
value="<?=$s?>" />
  </td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>
    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
  </td>
</tr>
</table>
</form>
</body>
</html>

```

Problema n.º 2

Enunciado: Hallar el cociente y el residuo (resto) de dos números enteros.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese dos números enteros; luego, que el sistema realice el cálculo respectivo para hallar el cociente y residuo. Para esto use la siguiente expresión:

Expresión algorítmica

$c \leftarrow n1 / n2$

$r \leftarrow n1 \text{ Mod } n2$

Entrada

- Dos números ($n1$ y $n2$)

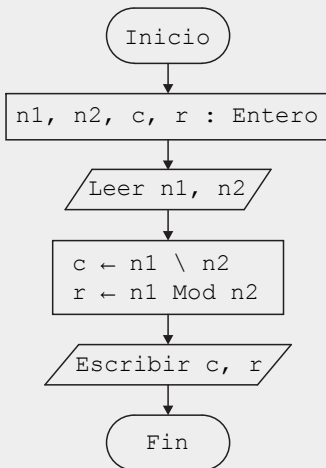
Salida

- El cociente (c)
- El residuo (r)

Diseño: Interfaz de usuario

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

n1, n2, c, r : Entero

//Entrada

Leer n1, n2

//Proceso

c ← n1 \ n2

r ← n1 Mod n2

//Salida

Escribir c, r

Fin

Codificación:

```

<?php

//Variables
$n1 = 0; $n2 = 0; $c = 0; $r = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n1 = (int)$_POST["txtn1"];
    $n2 = (int)$_POST["txtn2"];

    //Proceso
    $c = (int)($n1 / $n2);
    $r = $n1 % $n2;
  }
  
```

```

}
?>
<html>
<head>
<title>Problema 02</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema02.php">
    <table width="241" border="0">
        <tr>
            <td colspan="2"><strong>Problema 02</strong> </td>
        </tr>
        <tr>
            <td width="81">Numero 1 </td>
            <td width="150">
                <input name="txtn1" type="text" id="txtn1" value="<?=$n1?>" />
            </td>
        </tr>
        <tr>
            <td>Numero 2 </td>
            <td>
                <input name="txtn2" type="text" id="txtn2" value="<?=$n2?>" />
            </td>
        </tr>
        <tr>
            <td>Cociente</td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>"></td>
        </tr>
        <tr>
            <td>Residuo</td>
            <td>
                <input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>"></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" /></td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 3

Enunciado: Dado el valor de venta de un producto, hallar el IGV (19 %) y el precio de venta.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese el valor de venta del producto; luego, que el sistema realice el cálculo respectivo para hallar el IGV y el precio de venta. Para esto use la siguiente expresión.

Expresión algorítmica

$$\text{igv} \leftarrow \text{vv} * 0.19$$

$$\text{pv} \leftarrow \text{vv} + \text{igv}$$
Entrada

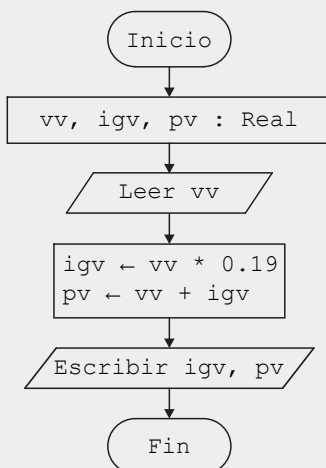
- Valor de venta (vv)

Salida

- El IGV (igv)
- El Precio de Venta (pv)

Diseño:**Interfaz de usuario**

The screenshot shows a web browser window titled 'Problema 03 - Windows Internet Explorer'. The address bar shows 'http://localhost/Cap02/Problema03.php'. The browser has a menu bar with 'Archivo', 'Edición', 'Ver', 'Favoritos', 'Herramientas', and 'Ayud'. Below the menu is a toolbar with a star icon and a plus icon. The main content area is titled 'Problema 03' and contains three input fields: 'Valor de venta' with the value '100', 'IGV' with the value '19', and 'Precio venta' with the value '119'. Below these fields is a button labeled 'Calcular'.

Diagrama de flujo**Algoritmo****Pseudocódigo****Inicio****//Variables**

`vv, igv, pv : Real`

//Entrada

`Leer vv`

//Proceso

`igv ← vv * 0.19`

`pv ← vv + igv`

//Salida

`Escribir igv, pv`

Fin

Codificación:

```

<?php

//Variables
$vv = 0.0; $igv = 0.0; $pv = 0.0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $vv = (float)$_POST["txtvv"];

    //Proceso
    $igv = $vv * 0.19;
    $pv = $vv + $igv;
}
?>
<html>
<head>
<title>Problema 03</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema03.php">
    <table width="270" border="0">
        <tr>
            <td colspan="2"><strong>Problema 03 </strong></td>
        </tr>
        <tr>
            <td width="109">Valor de venta </td>
            <td width="151">
                <input name="txtvv" type="text" id="txtvv" value="<?=$vv?>" />
            </td>
        </tr>
        <tr>
            <td>IGV</td>
            <td>
                <input name="txtigv" type="text" class="TextoFondo" id="txtigv"
                value="<?=$igv?>" /></td>
        </tr>
        <tr>
            <td>Precio venta </td>
            <td><input name="txtpv" type="text" class="TextoFondo" id="txtpv"
            value="<?=$pv?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
                value="Calcular" /></td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 4

Enunciado: Hallar la potencia de a^n , donde «a» y «n» pertenecen a $\mathbb{Z}^+$ (números enteros positivos).

Análisis: Para la solución de este problema, se requiere que el usuario ingrese dos números enteros positivos «a» y «n»; luego, que el sistema procese y obtenga la potencia «p».

Expresión matemática

$$p = a^n = \underbrace{a \times a \times a \times \dots \times a}_{n \text{ factores}}$$

Expresión algorítmica

$$p \leftarrow a^n$$

Entrada

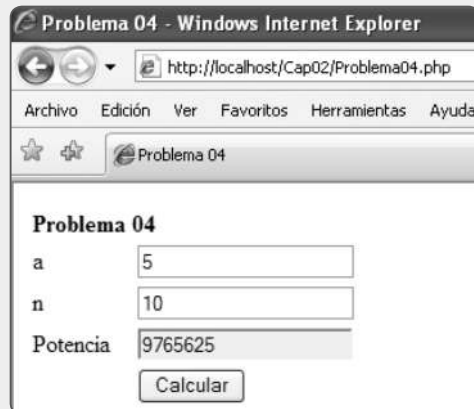
- Dos números enteros (a, n)

Salida

- La potencia (p)

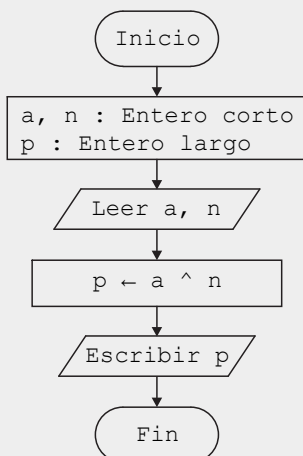
Diseño:

Interfaz de usuario



Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

a, n : Entero corto
p : Entero largo

//Entrada

Leer a, n

//Proceso

$p \leftarrow a^n$

//Salida

Escribir p

Fin

Codificación:

```

<?php

//Variables
$a = 0; $n = 0; $p = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $a = (int)$_POST["txta"];
    $n = (int)$_POST["txtn"];

    //Proceso
    $p = pow($a , $n);
}
?>

<html>
<head>
<title>Problema 04</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema04.php">
    <table width="226" border="0">
        <tr>
            <td colspan="2"><strong>Problema 04 </strong></td>
        </tr>
        <tr>
            <td width="67">a </td>
            <td width="149">
                <input name="txta" type="text" id="txta" value="<?=$a?>" />
            </td>
        </tr>
        <tr>
            <td>n</td>
            <td>
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Potencia</td>
            <td><input name="txtp" type="text" class="TextoFondo" id="txtp"
value="<?=$p?>"></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 5

Enunciado: Hallar la radicación de $\sqrt[n]{a}$, donde «a» y «n» pertenecen a $\mathbb{Z}^+$ (números enteros positivos).

Análisis: Para la solución de este problema, se requiere que el usuario ingrese dos números enteros positivos «a» y «n»; luego, que el sistema procese y obtenga la radicación «r».

Expresión matemática

$$r = \sqrt[n]{a} = a^{\frac{1}{n}}$$

Expresión algorítmica

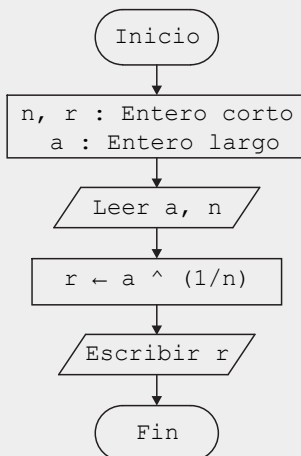
$$r \leftarrow a^{(1/n)}$$

Entrada

- Dos números enteros (a, n)

Salida

- La radicación (r)

Diseño:**Interfaz de usuario**
Diagrama de flujo**Algoritmo****Pseudocódigo****Inicio****//Variables**

n, r : Entero corto
a : Entero largo

//Entrada

Leer a, n

//Proceso

$r \leftarrow a^{(1/n)}$

//Salida

Escribir r

Fin

Codificación:

```

<?php

//Variables
$a = 0; $n = 0; $r = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $a = (int)$_POST["txta"];
    $n = (int)$_POST["txtn"];

    //Proceso
    $r = pow($a , (1/$n));
}
?>
<html>
<head>
<title>Problema 05</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema05.php">
    <table width="226" border="0">
        <tr>
            <td colspan="2"><strong>Problema 05</strong> </td>
        </tr>
        <tr>
            <td width="67">a </td>
            <td width="149">
                <input name="txta" type="text" id="txta" value="<?=$a?>" />
            </td>
        </tr>
        <tr>
            <td>n</td>
            <td>
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Radicacion</td>
            <td><input name="txtp" type="text" class="TextoFondo" id="txtp"
value="<?=$r?>"></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 6

Enunciado: Dado un número de 5 dígitos, devolver el número en orden inverso.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese un número «n»; luego, que el sistema procese y obtenga el número inverso «ni», realizando 4 divisiones sucesivas entre 10, para acumular el residuo y el último cociente.

$$\begin{array}{r}
 12345 \overline{)10} \\
 \underline{5} \\
 1234 \overline{)10} \\
 \underline{4} \\
 123 \overline{)10} \\
 \underline{3} \\
 12 \overline{)10} \\
 \underline{2} \\
 10 \overline{)10} \\
 \underline{1}
 \end{array}$$

Entrada

- Un número entero (n)

Salida

- El número inverso (ni)

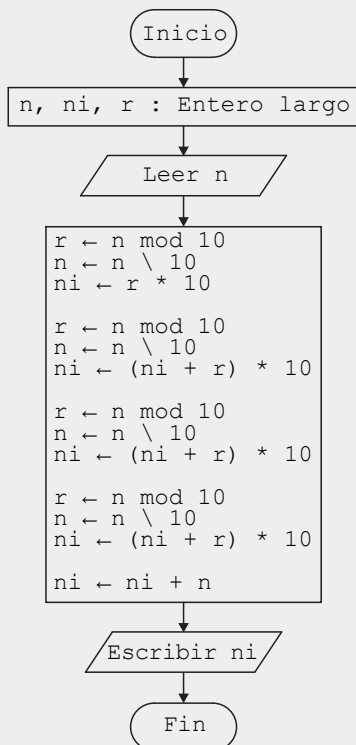
Diseño:

Interfaz de usuario



Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

```

//Variables
n, ni, r : Entero largo

//Entrada
Leer n

//Proceso
r ← n mod 10
n ← n \ 10
ni ← r * 10

r ← n mod 10
n ← n \ 10
ni ← (ni + r) * 10

r ← n mod 10
n ← n \ 10
ni ← (ni + r) * 10

r ← n mod 10
n ← n \ 10
ni ← (ni + r) * 10

r ← n mod 10
n ← n \ 10
ni ← (ni + r) * 10

r ← n mod 10
n ← n \ 10
ni ← ni + n

//Salida
Escribir ni
  
```

Fin

Codificación:

```

<?php
//Variables
$n = 0; $ni = 0; $r = 0; $tmp = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $tmp = $n;
    $r = $n % 10;
    $n = (int)($n / 10);
    $ni = $r * 10;

    $r = $n % 10;
    $n = (int)($n / 10);
    $ni = ($ni + $r) * 10;

    $r = $n % 10;
    $n = (int)($n / 10);
    $ni = ($ni + $r) * 10;

    $r = $n % 10;
    $n = (int)($n / 10);
    $ni = ($ni + $r) * 10;

    $ni = $ni + $n;
    $n = $tmp;
}
?>
<html>
<head>
<title>Problema 06</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema06.php">
    <table width="237" border="0">
        <tr>
            <td colspan="2"><strong>Problema 06</strong> </td>
        </tr>
        <tr>
            <td width="83">Numero</td>
            <td width="144">
                <input name="txtn" type="text" id="txtn" value="<?=$n?" />
            </td>
        </tr>
    </table>

```

```

<tr>
  <td>Inverso</td>
  <td>
    <input name="txtni" type="text" class="TextoFondo" id="txtni"
value="<?=$ni?>" />
  </td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>
    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
  </td>
</tr>
</table>
</form>
</body>
</html>

```

Problema n.º 7

Enunciado: Determinar la suma de los N primeros números enteros positivos (Z^+) use la siguiente fórmula.

$$S = \frac{N(N+1)}{2}$$

Análisis: Para la solución de este problema, se requiere que el usuario ingrese un número entero positivo «n»; luego, que el sistema procese y obtenga la suma de los primeros números enteros positivos hasta «n».

Expresión matemática

$$S = \frac{N(N+1)}{2}$$

Expresión algorítmica

$s \leftarrow (n * (n + 1)) / 2$

Entrada

- Número entero (n).

Salida

- Suma (s).

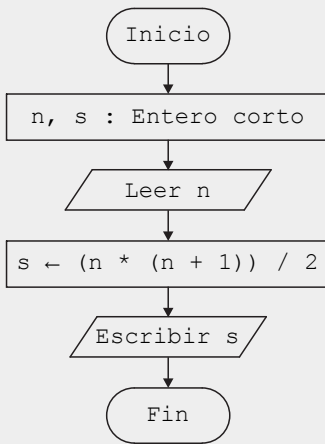
Diseño:

Interfaz de usuario

The screenshot shows a web browser window titled "Problema 07 - Windows Internet Explorer". The address bar displays the URL "http://localhost/Cap02/Problema07.php". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The page content is titled "Problema 07" and contains a form with two input fields: "Numero" with the value "5" and "Suma" with the value "15". Below these fields is a "Calcular" button.

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

n, s : Entero corto

//Entrada

Leer n

//Proceso

$s \leftarrow (n * (n + 1)) / 2$

//Salida

Escribir s

Fin

Codificación:

```

<?php

//Variables
$n = 0; $s = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $s = ($n * ($n + 1)) / 2;
}
?>
<html>
<head>
<title>Problema 07</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema07.php">
    <table width="237" border="0">
        <tr>
            <td colspan="2"><strong>Problema 07</strong> </td>
        </tr>
        <tr>

```

```
<td width="83">Numero</td>
<td width="144">
    <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
</td>
</tr>
<tr>
<td>Suma</td>
<td>
    <input name="txtni" type="text" class="TextoFondo" id="txtni"
value="<?=$s?>" />
</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>
    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
</td>
</tr>
</table>
</form>
</body>
</html>
```

Problema n.º 8

Enunciado: Calcular el interés compuesto, generado por un capital depositado durante cierta cantidad de tiempo a una tasa de interés determinada; aplique las siguientes fórmulas:

$$M = (1 + r \%)^t \cdot C$$

$$I = M - C$$

Monto (M): Es la suma del capital más los intereses producidos en determinado tiempo.

Tasa de interés (r %): Es la ganancia que se obtiene por cada 100 unidades monetarias en cada periodo de tiempo.

Capital (C): Es todo aquello que se va a ceder o imponer durante algún tiempo para generar una ganancia.

Interés (I): Parte de la utilidad que obtiene el capitalista al prestar su dinero.

Tiempo (t): Es el periodo de tiempo durante el cual se cede el capital.

Análisis: Para la solución de este problema se requiere que el usuario ingrese el capital «c» y la tasa de interés «r»; luego, que el sistema procese y obtenga el interés ganado y el monto producido.

Expresión matemática

$$M = (1 + r\%)^t \cdot C$$

Expresión algorítmica

$$m \leftarrow ((1 + r / 100) ^ t) * c$$

Entrada

- Capital (c)
- Tasa de interés (r)
- Tiempo (t)

Salida

- Interés (i)
- Monto (m)

Diseño:**Interfaz de usuario**

Problema 08 - Windows Internet Explorer

http://localhost/Cap02/Problema08.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 08

Problema 08

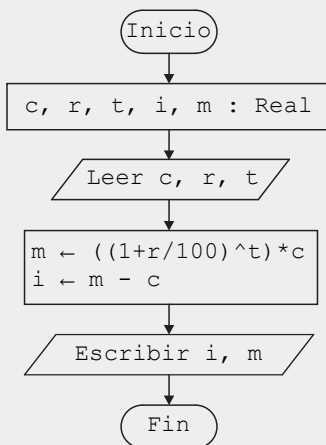
Capital

Tasa de interes

Tiempo

Intereses

Monto

Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio****//Variables** $c, r, t, i, m : \text{Real}$ **//Entrada**Leer c, r, t **//Proceso** $m \leftarrow ((1+r/100)^t) * c$ $i \leftarrow m - c$ **//Salida**Escribir i, m **Fin****Codificación:**

```

<?php

//Variables
$c = 0.0; $r = 0.0; $t = 0.0; $i = 0.0; $m = 0.0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $c = (float)$_POST["txtc"];
    $r = (float)$_POST["txtr"];
    $t = (float)$_POST["txtt"];

    //Proceso
    $m = pow((1 + $r / 100), $t) * $c;
    $i = $m - $c;
  }

```

```

}
?>
<html>
<head>
<title>Problema 08</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema08.php">
    <table width="264" border="0">
        <tr>
            <td colspan="2"><strong>Problema 08</strong> </td>
        </tr>
        <tr>
            <td width="83">Capital</td>
            <td width="144">
                <input name="txtc" type="text" id="txtc" value="<?=$c?>" />
            </td>
        </tr>
        <tr>
            <td>Tasa de interes </td>
            <td>
                <input name="txtr" type="text" id="txtr" value="<?=$r?>" />
            </td>
        </tr>
        <tr>
            <td>Tiempo</td>
            <td><input name="txtt" type="text" id="txtt" value="<?=$t?>"></td>
        </tr>
        <tr>
            <td>Intereses</td>
            <td><input name="txti" type="text" class="TextoFondo" id="txti" value="<?=$i?>"></td>
        </tr>
        <tr>
            <td>Monto</td>
            <td><input name="txtm" type="text" class="TextoFondo" id="txtm" value="<?=$m?>"></td>
        </tr>
        <tr>
            <td>&nbsp;  </td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular" value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 9

Enunciado: Crear un programa para encontrar el área de un círculo, use la fórmula:

$$A = \pi \cdot r^2$$

Área (A): Es el área del círculo.

PI (π): Representa el valor constante pi (3.14159)

Radio (r): Es el radio del círculo

Análisis: Para la solución de este problema se requiere que el usuario ingrese el radio del círculo; luego, que el sistema procese y obtenga el área del círculo.

Expresión aritmética

$$A = \pi \cdot r^2$$

Expresión algorítmica

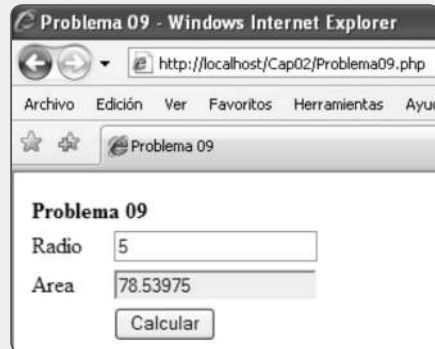
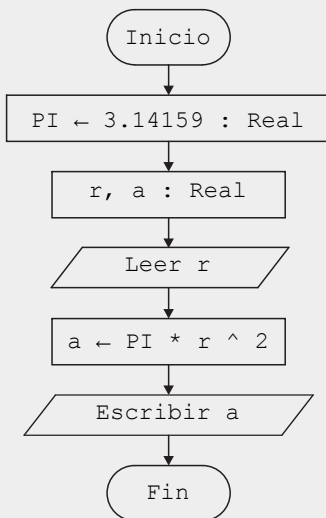
$A \leftarrow 3.14159 * r^2$

Entrada

- Radio (r)

Salida

- Área (a)

Diseño:**Interfaz de usuario****Diagrama de flujo****Algoritmo****Inicio**

//Constantes

PI = 3.14159 : Real

//Variables

r, a : Real

//Entrada

Leer r

//Proceso

$a \leftarrow PI * r^2$

//Salida

Escribir a

Fin**Pseudocódigo**

Codificación:

```

<?php
//Constantes
define(PI,3.14159);

//Variables
$a = 0; $r = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $r = (float)$_POST["txtr"];

    //Proceso
    $a = PI * pow($r, 2);
}
?>
<html>
<head>
<title>Problema 09</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema09.php">
    <table width="217" border="0">
        <tr>
            <td colspan="2"><strong>Problema 09</strong> </td>
        </tr>
        <tr>
            <td width="55">Radio</td>
            <td width="152">
                <input name="txtr" type="text" id="txtr" value="<?=$r?>" />
            </td>
        </tr>
        <tr>
            <td>Area</td>
            <td>
                <input name="txta" type="text" class="TextoFondo" id="txta"
value="<?=$a?>"/>
            </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 10

Enunciado: Crear un programa que permita convertir una cantidad de segundos en horas, minutos y segundos.

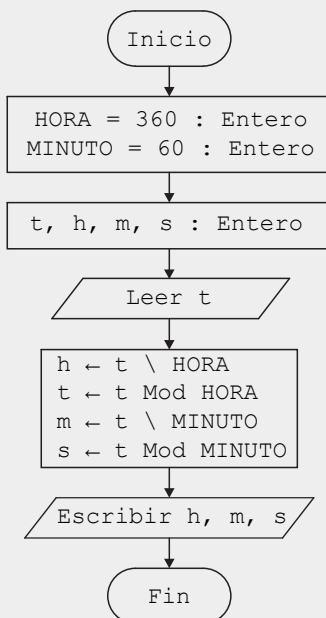
Análisis: Para la solución de este problema se requiere que el usuario ingrese un tiempo expresado en segundos; luego, que el sistema procese y obtenga las horas, minutos y segundos restantes.

Entrada

- Tiempo en segundos (t)

Salida

- Horas (h)
- Minutos (m)
- Segundos (s)

Diseño:**Interfaz de usuario**
Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio****//Constantes**

```

HORA = 360 : Entero
MINUTO = 60 : Entero

```

//Variables

```

t, h, m, s : Entero

```

//Entrada

```

Leer t

```

//Proceso

```

h ← t \ HORA
t ← t Mod HORA
m ← t \ MINUTO
s ← t Mod MINUTO

```

//Salida

```

Escribir h, m, s

```

Fin

Codificación:

```

<?php
//Constantes
define (HORA, 3600);
define (MINUTO, 60);

//Variables
$t = 0; $h = 0; $m = 0; $s = 0; $tt = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $t = $_POST["txtt"];

    //Proceso
    $tt = $t;
    $h = (int)($t / HORA);
    $t = $t % HORA;
    $m = (int)($t / MINUTO);
    $s = $t % MINUTO;
    $t = $tt;
}
?>
<html>
<head>
<title>Problema 10</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema10.php">
    <table width="290" border="0">
        <tr>
            <td colspan="2"><strong>Problema 10</strong> </td>
        </tr>
        <tr>
            <td width="134">Tiempo en segundos </td>
            <td width="146">
                <input name="txtt" type="text" id="txtt" value="<?=$t?>" />
            </td>
        </tr>
        <tr>
            <td>Horas</td>
            <td>
                <input name="txth" type="text" class="TextoFondo" id="txth"
                value="<?=$h?>" />
            </td>
        </tr>
    </table>
</form>

```

```
</tr>
<tr>
  <td>Minutos</td>
  <td><input name="txtm" type="text" class="TextoFondo" id="txtm"
value="<?=$m?>"></td>
</tr>
<tr>
  <td>Segundos</td>
  <td><input name="txts" type="text" class="TextoFondo" id="txts"
value="<?=$s?>"></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>
    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
  </td>
</tr>
</table>
</form>
</body>
</html>
```

2.2 Problemas propuestos

Los siguientes problemas le servirán para medir su aprendizaje, es importante que los resuelva.

Propuesto n.º 1

Enunciado: Dado dos números enteros (Z), a y b , hallar $a + b$ y $a - b$.

Propuesto n.º 2

Enunciado: Dado dos números enteros, determinar cuántos números enteros están incluidos en ellos.

Propuesto n.º 3

Enunciado: Dada una cantidad de milímetros, expresarlo en la máxima cantidad de metros, el resto en decímetros, centímetros y milímetros.

Propuesto n.º 4

Enunciado: Obtener el valor de «c» y «d» de acuerdo a la siguiente fórmula.

$$c = \frac{(4a^4 + 3ba + b^2)}{a^2 - b^2} \quad d = \frac{(3c^2 + a + b)}{4}$$

Propuesto n.º 5

Enunciado: Dado 4 números enteros, obtener el porcentaje de cada uno en función a la suma de los 4 números ingresados.

Propuesto n.º 6

Enunciado: Hallar el área y el perímetro de un cuadrado.

Propuesto n.º 7

Enunciado: Dada una cantidad de horas obtener su equivalente en minutos y segundos.

Propuesto n.º 8

Enunciado: Convertir una cantidad de grados Fahrenheit a Celsius y Kelvin.

Propuesto n.º 9

Enunciado: Hallar el área y el perímetro de un rectángulo.

Propuesto n.º 10

Enunciado: Convertir grados sexagesimales a centesimales.

Estructura selectiva simple y doble

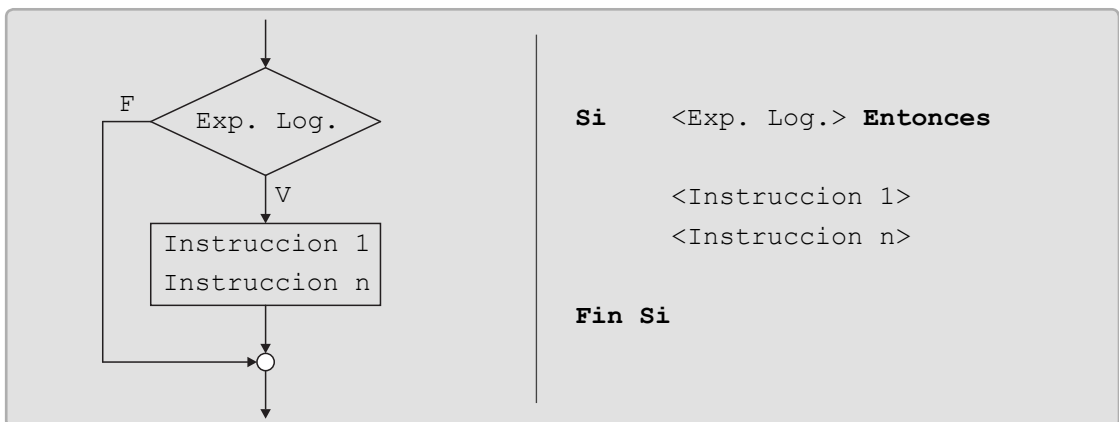
3.1 Introducción

Muchas veces tenemos que decidir si realizar una u otra tarea, dependiendo de una condición; en la programación existe una estructura que permite evaluar una condición (expresión lógica que devuelve «verdadero» o «falso») y determina qué instrucción o instrucciones se deben ejecutar si la condición es verdadera o si la condición es falsa.

En este capítulo usted aprenderá a resolver problemas que permitan evaluar condiciones lógicas; esta es una de las estructuras básicas y más utilizadas en todo lenguaje de programación, también se las conoce como estructuras condicionales, alternativas y de decisiones.

3.2 Estructura selectiva simple

Evalúa una expresión lógica (condición), si es verdadera ejecuta una determinada instrucción o instrucciones.

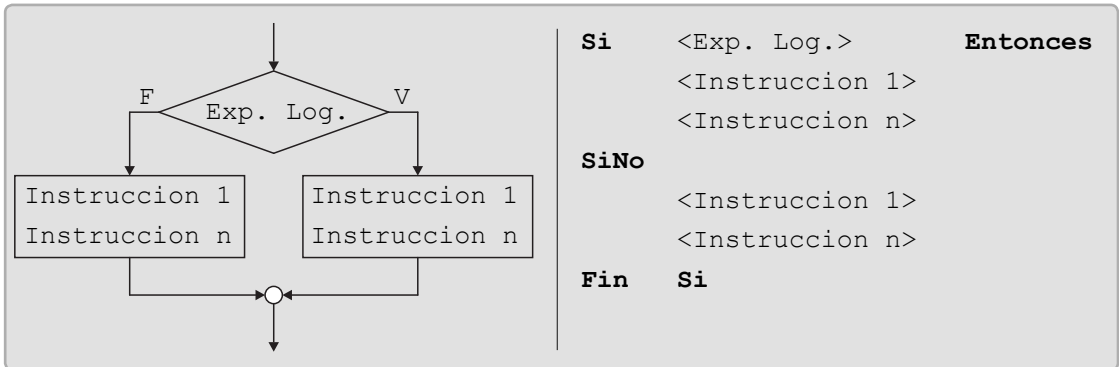


Sintaxis PHP

```
//Una instrucción  
if (<Exp. Log.>)  
    <Instruccion 1>;  
  
//Varias instrucciones  
if (<Exp. Log.>) {  
    <Instruccion 1>;  
    <Instruccion n>;  
}
```

3.3 Estructura selectiva doble

Evalúa una expresión lógica (condición), si es verdadero ejecuta una o varias instrucciones y si es falso ejecuta otro grupo de instrucciones.



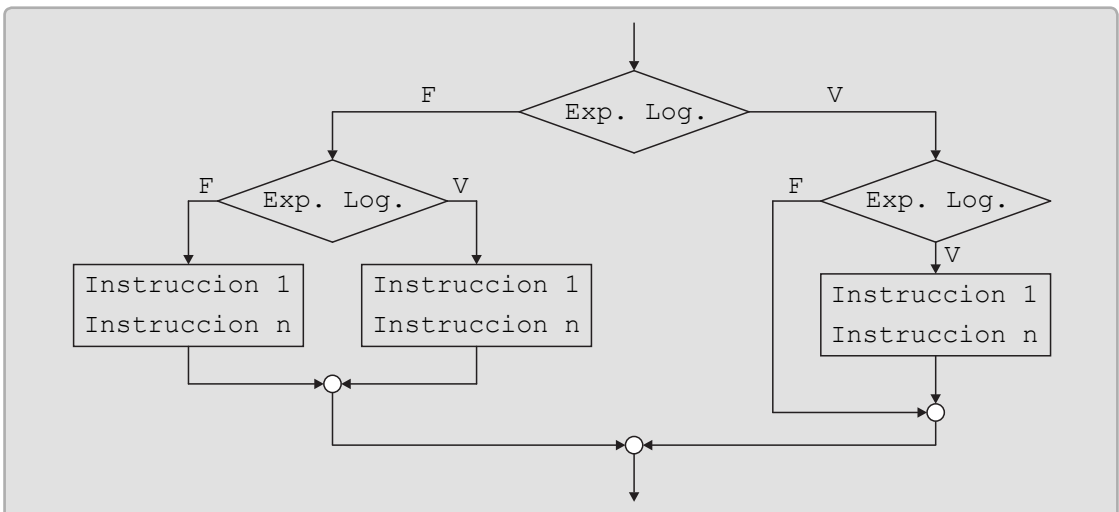
Sintaxis PHP

```

if <Exp. Log.> {
    <Instruccion 1>;
    <Instruccion n>;
} else {
    <Instruccion 1>;
    <Instruccion n>;
}
  
```

3.4 Estructuras anidadas

Son aquellas estructuras que contienen una o más estructuras; es decir, está permitido colocar dentro de una estructura otra estructura.



```

Si <Exp. Log.> Entonces
  Si <Exp. Log.> Entonces
    <Instruccion 1>
    <Instruccion n>
  Fin Si
SiNo
  Si <Exp. Log.> Entonces
    <Instruccion 1>
    <Instruccion n>
  SiNo
    <Instruccion 1>
    <Instruccion n>
  Fin Si
Fin Si

```

Sintaxis PHP

```

if (<Exp. Log.>) {
  if (<Exp. Log.>) {
    <Instruccion 1>;
    <Instruccion n>;
  }
} else {
  if (<Exp. Log.>) {
    <Instruccion 1>;
    <Instruccion n>;
  } else {
    <Instruccion 1>;
    <Instruccion n>;
  }
}

```

Problema n.º 11

Enunciado: Dado dos números enteros diferentes, devolver el número mayor.

Análisis: Para la solución de este problema se requiere que el usuario ingrese dos números enteros diferentes; luego, que el sistema realice el proceso para devolver el número mayor.

Expresión

Si $n1 > n2 \Rightarrow n1$ es mayor

Si $n2 > n1 \Rightarrow n2$ es mayor

Entrada

- Dos números ($n1$ y $n2$)

Salida

- Número mayor (m)

Diseño:
Interfaz de usuario

Problema 11 - Windows Internet Explorer

http://localhost/Cap03/Problema11.php

Archivo Edición Ver Favoritos Herramientas Ayu

Problema 11

Problema 11

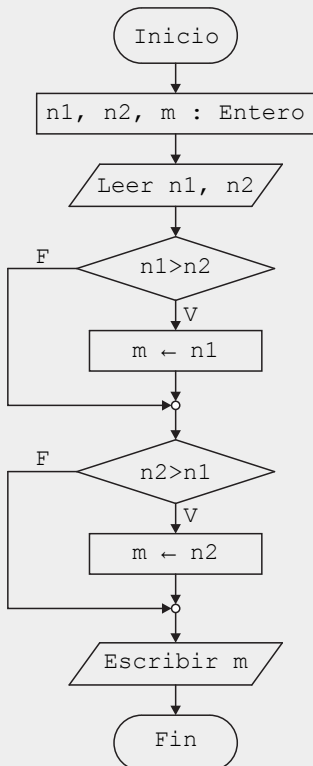
Numero 1

Numero 2

Mayor

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

n1, n2, m : Entero

//Entrada

Leer n1, n2

//Proceso

Si n1 > n2 Entonces

m ← n1

Fin Si

Si n2 > n1 Entonces

m ← n2

Fin Si

//Salida

Escribir m

Fin

Codificación:

```

<?php

//Variables
$n1 = 0; $n2 = 0; $m = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n1 = $_POST["txtn1"];
    $n2 = $_POST["txtn2"];

    //Proceso
    if($n1 > $n2)
        $m = $n1;

    if($n2 > $n1)
        $m = $n2;
}
?>
<html>
<head>
<title>Problema 11</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema11.php">
    <table width="236" border="0">
        <tr>
            <td colspan="2"><strong>Problema 11</strong> </td>
        </tr>
        <tr>
            <td width="75">Numero 1 </td>
            <td width="151">
                <input name="txtn1" type="text" id="txtn1" value="<?=$n1?>" />
            </td>
        </tr>
        <tr>
            <td>Numero 2 </td>
            <td>
                <input name="txtn2" type="text" id="txtn2" value="<?=$n2?>" />
            </td>
        </tr>
        <tr>
            <td>Mayor</td>

```

```

        <td><input name="txtm" type="text" class="TextoFondo" id="txtm"
value="<?=$m?>"></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 12

Enunciado: Determinar si un número entero es positivo, negativo o neutro.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número entero; luego, que el sistema verifique si es «positivo», «negativo» o «neutro».

Expresión

Si $n > 0 \Rightarrow$ POSITIVO

Si $n < 0 \Rightarrow$ NEGATIVO

Si $n = 0 \Rightarrow$ NEUTRO

Entrada

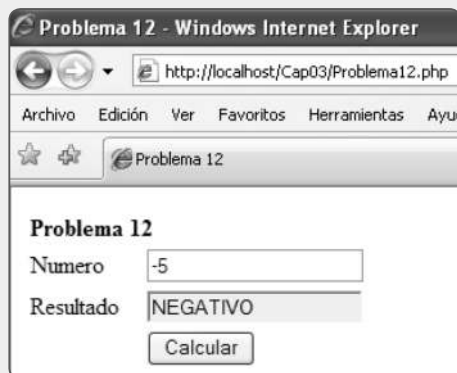
- Número (n)

Salida

- Resultado (r)
 - POSITIVO
 - NEGATIVO
 - NEUTRO

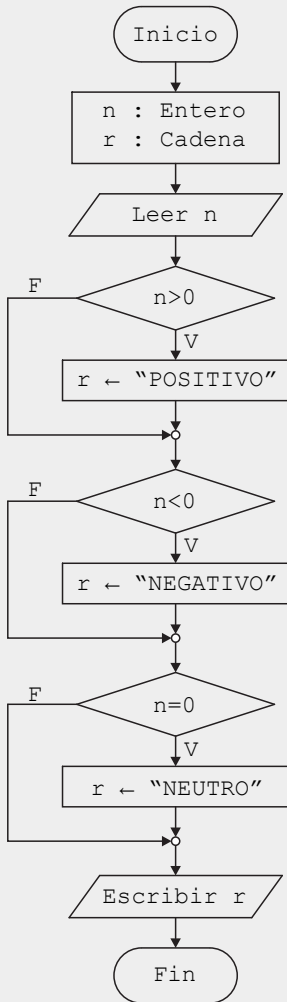
Diseño:

Interfaz de usuario



Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

n : Entero

r : Cadena

//Entrada

Leer n

//Proceso

Si n > 0 Entonces

r ← "POSITIVO"

Fin Si

Si n < 0 Entonces

r ← "NEGATIVO"

Fin Si

Si n = 0 Entonces

r ← "NEUTRO"

Fin Si

//Salida

Escribir r

Fin

Codificación:

```

<?php

//Variables
$n = 0; $r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = $_POST["txtn"];

    //Proceso
    if($n > 0) {
        $r = "POSITIVO";
    }
}

```

```
        if($n < 0){
            $r = "NEGATIVO";
        }

        if($n == 0){
            $r = "NEUTRO";
        }
    }
?>
<html>
<head>
<title>Problema 12</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema12.php">
    <table width="236" border="0">
        <tr>
            <td colspan="2"><strong>Problema 12</strong> </td>
        </tr>
        <tr>
            <td width="75">Numero </td>
            <td width="151">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Resultado </td>
            <td>
                <input name="txtr" type="text" class="TextoFondo" id="txtr"
                value="<?=$r?>" />
            </td>
        </tr>
        <tr>
            <td>&nbsp;  </td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
                value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>
```

Problema n.º 13

Enunciado: Dado un carácter, determinar si es una vocal.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese un carácter y el sistema verifica si es una vocal.

Expresión

Si $c = 'a' \vee c = 'A' \Rightarrow \text{VOCAL}$

Si $c = 'e' \vee c = 'E' \Rightarrow \text{VOCAL}$

Si $c = 'i' \vee c = 'I' \Rightarrow \text{VOCAL}$

Si $c = 'o' \vee c = 'O' \Rightarrow \text{VOCAL}$

Si $c = 'u' \vee c = 'U' \Rightarrow \text{VOCAL}$

Entrada

- Carácter (c)

Salida

- Resultado (r)
 - ES VOCAL
 - NO ES VOCAL

Diseño:**Interfaz de usuario**

Problema 13 - Windows Internet Explorer

http://localhost/Cap03/Problema13.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 13

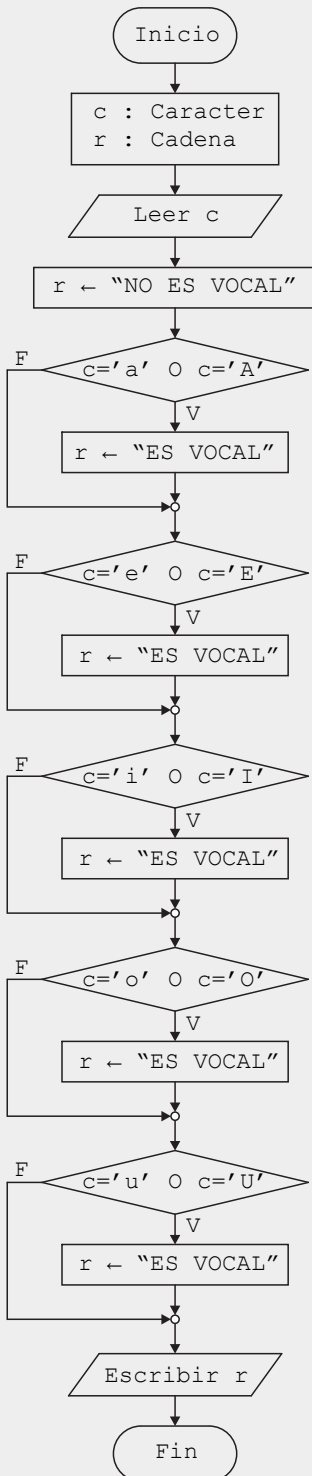
Problema 13

Caracter

Resultado

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

c : Caracter

r : Cadena

//Entrada

Leer c

//Proceso

r ← "NO ES VOCAL"

Si c='a' O c='A' Entonces
 r ← "ES VOCAL"

Fin Si

Si c='e' O c='E' Entonces
 r ← "ES VOCAL"

Fin Si

Si c='i' O c='I' Entonces
 r ← "ES VOCAL"

Fin Si

Si c='o' O c='O' Entonces
 r ← "ES VOCAL"

Fin Si

Si c='u' O c='U' Entonces
 r ← "ES VOCAL"

Fin Si

//Salida

Escribir r

Fin

Codificación:

```

<?php
//Variables
$c = ""; $r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $c = $_POST["txtc"];

    //Proceso
    $r = "NO ES VOCAL";

    if($c == "a" || $c == "A") {
        $r = "ES VOCAL";
    }

    if($c == "e" || $c == "E") {
        $r = "ES VOCAL";
    }

    if($c == "i" || $c == "I") {
        $r = "ES VOCAL";
    }

    if($c == "o" || $c == "O") {
        $r = "ES VOCAL";
    }

    if($c == "u" || $c == "U") {
        $r = "ES VOCAL";
    }
}
?>
<html>
<head>
<title>Problema 13</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema13.php">
    <table width="236" border="0">
        <tr>
            <td colspan="2"><strong>Problema 13</strong> </td>
        </tr>
        <tr>
            <td width="75">Caracter</td>
            <td width="151">

```

```

        <input name="txtc" type="text" id="txtc" value="<?=$c?>" />
</td>
</tr>
<tr>
    <td>Resultado </td>
    <td>
        <input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" />
    </td>
</tr>
<tr>
    <td>&nbsp;  </td>
    <td>
        <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
    </td>
</tr>
</table>
</form>
</body>
</html>

```

Problema n.º 14

Enunciado: Determinar si un número es múltiplo de 3 y 5.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número entero n ; luego, que el sistema analice y determine si el número es múltiplo de 3 y de 5.

Expresión

Si $n \text{ Mod } 3 = 0 \wedge n \text{ Mod } 5 = 0 \Rightarrow$

SI ES MULTIPLIO DE 3 y 5

SiNo

NO ES MULTIPLIO DE 3 y 5

Entrada

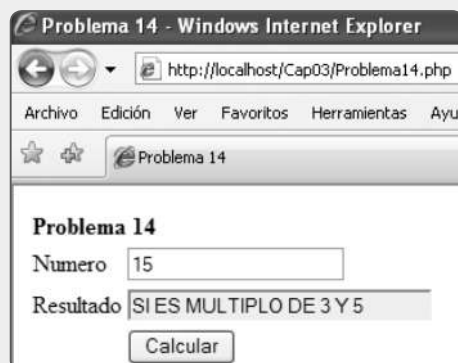
- Número (n)

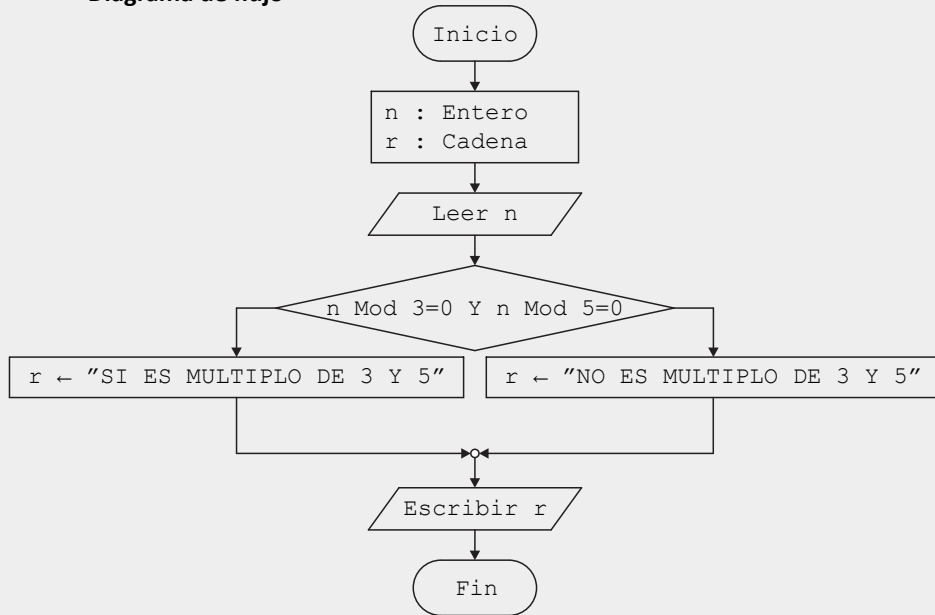
Salida

- Resultado (r)
 - ES MULTIPLIO
 - NO ES MULTIPLIO

Diseño:

Interfaz de usuario



Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio****//Variables**

n : Entero

r : Cadena

//Entrada

Leer n

//ProcesoSi $n \text{ Mod } 3 = 0$ Y $n \text{ Mod } 5 = 0$ Entonces $r \leftarrow \text{"SI ES MULTIPLO DE 3 y 5"}$

SiNo

 $r \leftarrow \text{"NO ES MULTIPLO DE 3 y 5"}$

Fin Si

//Salida

Escribir r

Fin

Codificación:

```

<?php

//Variables
$n = 0; $r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = $_POST["txtn"];

    //Proceso
    if($n % 3 == 0 && $n % 5 == 0) {
        $r = "SI ES MULTIPLO DE 3 Y 5";
    }else{
        $r = "NO ES MULTIPLO DE 3 Y 5";
    }
}
?>
<html>
<head>
<title>Problema 14</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema14.php">
    <table width="258" border="0">
        <tr>
            <td colspan="2"><strong>Problema 14</strong> </td>
        </tr>
        <tr>
            <td width="75">Numero</td>
            <td width="151">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Resultado </td>
            <td>
                <input name="txtr" type="text" class="TextoFondo" id="txtr"
                value="<?=$r?>" size="30"/>
            </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
                value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema 15

Enunciado: Determinar si un número entero es par o impar.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número entero n ; luego, que el sistema verifique si el número es par o impar.

Expresión

Si $n \text{ Mod } 2 = 0 \Rightarrow$

PAR

SiNo

IMPAR

Entrada

- Número (n)

Salida

- Resultado (r)
 - PAR
 - IMPAR

Diseño:

Interfaz de usuario

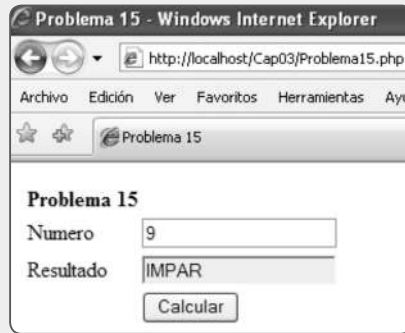
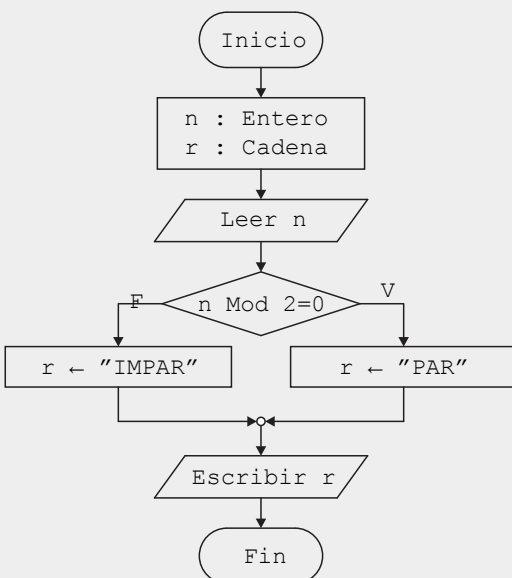


Diagrama de flujo



Algoritmo

Pseudocódigo

Inicio

//Variables

n : Entero
 r : Cadena

//Entrada

Leer n

//Proceso

Si $n \text{ Mod } 2 = 0$ Entonces
 $r \leftarrow \text{"PAR"}$
 SiNo
 $r \leftarrow \text{"IMPAR"}$
 Fin Si

//Salida

Escribir r

Fin

Codificación:

```

<?php

//Variables
$n = 0; $r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = $_POST["txtn"];

    //Proceso
    if($n % 2 == 0){
        $r = "PAR";
    }else{
        $r = "IMPAR";
    }
}
?>
<html>
<head>
<title>Problema 15</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema15.php">
    <table width="258" border="0">
        <tr>
            <td colspan="2"><strong>Problema 15</strong> </td>
        </tr>
        <tr>
            <td width="75">Numero</td>
            <td width="151">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Resultado </td>
            <td>
                <input name="txtr" type="text" class="TextoFondo" id="txtr"
                value="<?=$r?>" />
            </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
                value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 16

Enunciado: Dados tres números enteros, devolver el número mayor.

Análisis: Para la solución de este problema se requiere que el usuario ingrese tres números enteros (n_1 , n_2 y n_3); luego, que el sistema verifique y devuelva el número mayor.

Entrada

- Tres números (n_1 , n_2 , n_3)

Salida

- Número mayor (m)

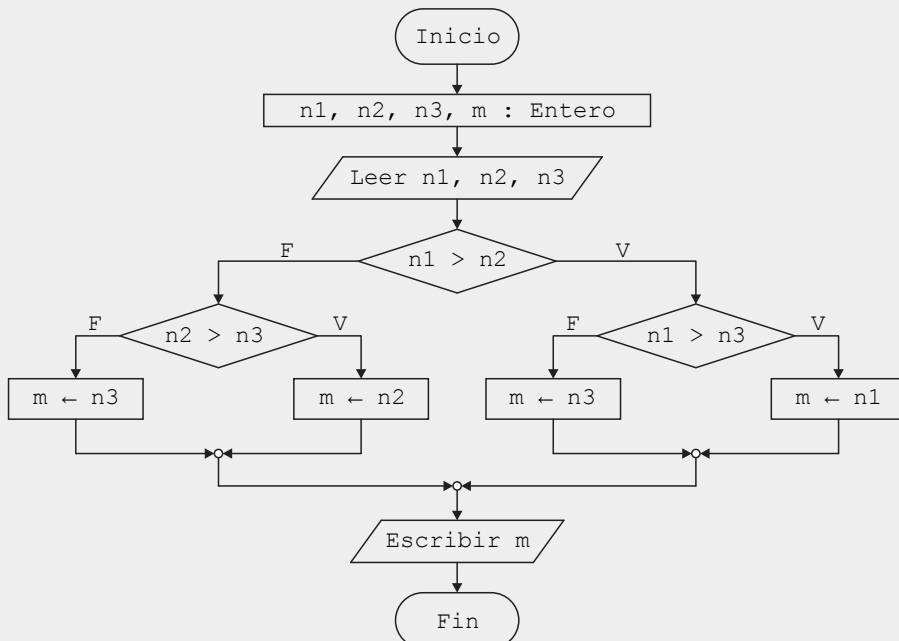
Diseño:

Interfaz de usuario

The screenshot shows a web browser window titled 'Problema 16 - Windows Internet Explorer'. The address bar shows 'http://localhost/Cap03/Problema16.php'. The browser menu includes 'Archivo', 'Edición', 'Ver', 'Favoritos', 'Herramientas', and 'Ayuda'. The page content is titled 'Problema 16' and contains four input fields: 'Numero 1' with the value 22, 'Numero 2' with the value 5, 'Numero 3' with the value 45, and 'Mayor' with the value 45. Below these fields is a 'Calcular' button.

Algoritmo

Diagrama de flujo



Pseudocódigo**Inicio**

```
//Variables
n1, n2, n3 : Entero

//Entrada
Leer n1, n2, n3

//Proceso
Si n1 > n2 Entonces
    Si n1 > n3 Entonces
        m ← n1
    SiNo
        m ← n3
    Fin Si
SiNo
    Si n2 > n3 Entonces
        m ← n2
    SiNo
        m ← n3
    Fin Si
Fin Si

//Salida
Escribir m
```

Fin**Codificación:**

```
<?php

//Variables
$n1 = 0; $n2 = 0; $n3 = 0; $m = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n1 = (int)$_POST["txtn1"];
    $n2 = (int)$_POST["txtn2"];
    $n3 = (int)$_POST["txtn3"];

    //Proceso
    if($n1 > $n2){
        if($n1 > $n3){
            $m = $n1;
        }else{
            $m = $n3;
        }
    }else{
        if($n2 > $n3){
            $m = $n2;
        }else{
            $m = $n3;
        }
    }
}
```

```

}
?>
<html>
<head>
<title>Problema 16</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema16.php">
    <table width="258" border="0">
        <tr>
            <td colspan="2"><strong>Problema 16</strong> </td>
        </tr>
        <tr>
            <td width="75">Numero 1 </td>
            <td width="151">
                <input name="txtn1" type="text" id="txtn1" value="<?=$n1?>" />
            </td>
        </tr>
        <tr>
            <td>Numero 2 </td>
            <td>
                <input name="txtn2" type="text" id="txtn2" value="<?=$n2?>" />
            </td>
        </tr>
        <tr>
            <td>Numero 3 </td>
            <td><input name="txtn3" type="text" id="txtn3" value="<?=$n3?>"></td>
        </tr>
        <tr>
            <td>Mayor</td>
            <td><input name="txtm" type="text" class="TextoFondo" id="txtm"
value="<?=$m?>"></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 17

Enunciado: Dado un número, devolver el doble si el número no es par; caso contrario, el triple.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número entero n ; luego, que el sistema verifique y devuelva el doble o el triple del número.

Expresión

Si $\sim(n \bmod 2 = 0) \Rightarrow$

$r = n * 2$

SiNo

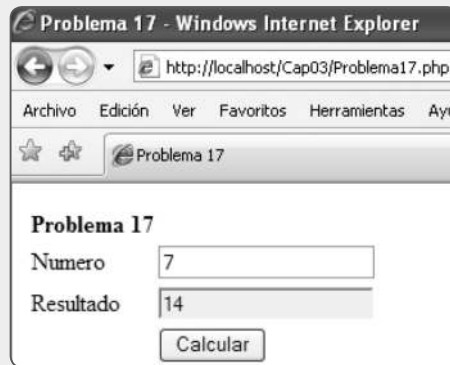
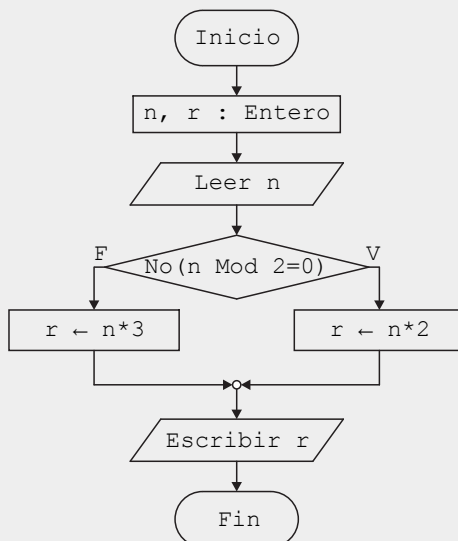
$r = n * 3$

Entrada

- Número entero (n)

Salida

- Resultado (r)

Diseño:**Interfaz de usuario****Algoritmo****Diagrama de flujo****Pseudocódigo****Inicio**

//Variables

$n, r : \text{Entero}$

//Entrada

Leer n

//Proceso

Si No($n \bmod 2=0$) Entonces

$r \leftarrow n * 2$

SiNo

$r \leftarrow n * 3$

Fin Si

//Salida

Escribir r

Fin

Codificación:

```
<?php

//Variables
$n = 0; $r = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    if(!($n % 2 == 0)){
        $r = $n * 2;
    }else{
        $r = $n * 3;
    }
}
?>

<html>
<head>
<title>Problema 17</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema17.php">
    <table width="258" border="0">
        <tr>
            <td colspan="2"><strong>Problema 17</strong> </td>
        </tr>
        <tr>
            <td width="75">Numero </td>
            <td width="151">
                <input name="txtn" type="text" id="txtn" value="<?=$n?" />
            </td>
        </tr>
        <tr>
            <td>Resultado</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?"></td>
        </tr>
    </table>
</form>
</body>
</html>
```

```
<tr>
  <td>&nbsp;</td>
  <td>
    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
  </td>
</tr>
</table>
</form>
</body>
</html>
```

Problema n.º 18

Enunciado: Dado que se tienen 3 números, devolver los números en orden ascendente.

Análisis: Para la solución de este problema se requiere que el usuario ingrese tres números (n1, n2 y n3); luego, que el sistema verifique y devuelva los números ordenados en forma ascendente.

Primero se debe encontrar el número mayor, luego el número menor y, al final, el número intermedio; este resulta de sumar los tres números y luego restar el resultado de (mayor + menor).

Entrada

- Números (n1, n2, n3)

Salida

- Números ordenados (ma, int, me)

Diseño:

Interfaz de usuario

Problema 18 - Windows Internet Explorer

http://localhost/Cap03/Problema18.php

Archivo Edición Ver Favoritos Herramientas Ayu

☆ ☆ Problema 18

Problema 18

Numero 1 5

Numero 2 22

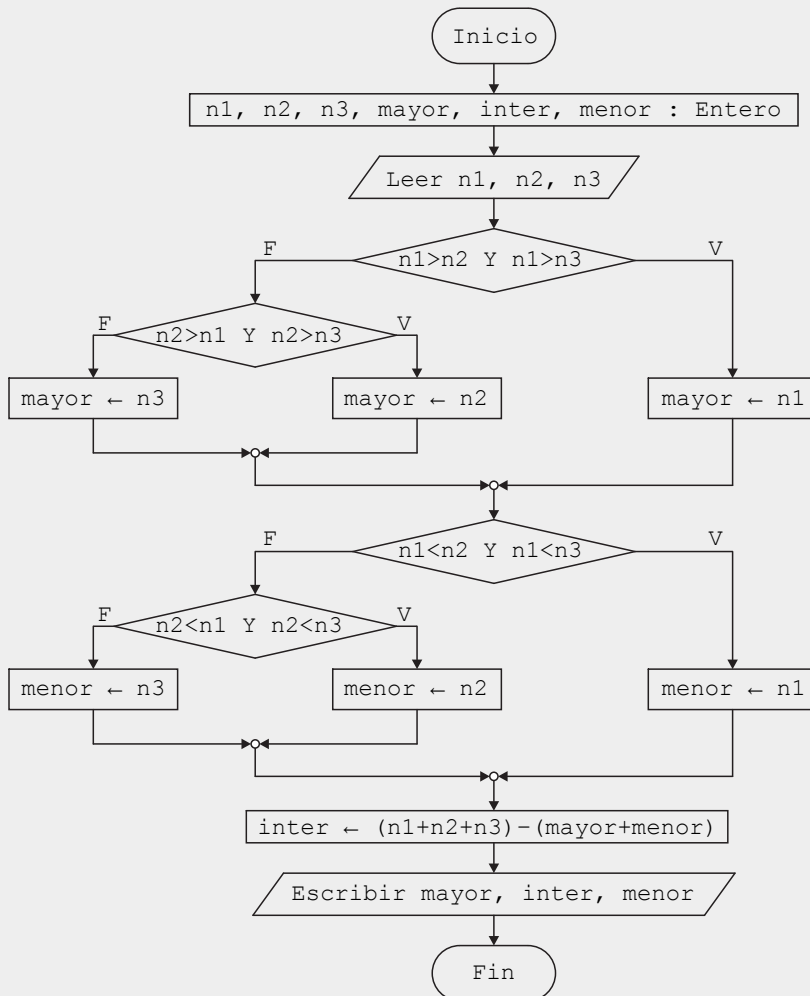
Numero 3 111

Mayor 111

Intermedio 22

Menor 5

Calcular

Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio****//Variables**

n1, n2, n3, mayor, inter, menor : Entero

//Entrada

Leer n1, n2, n3

//Proceso

Si n1 > n2 Y n1 > n3 Entonces

mayor ← n1

SiNo

Si n2 > n1 Y n2 > n3 Entonces

mayor ← n2

SiNo

mayor ← n3

Fin Si

Fin Si

```

Si n1 < n2 Y n1 < n3 Entonces
    menor ← n1
SiNo
    Si n2 < n1 Y n2 < n3 Entonces
        menor ← n2
    SiNo
        menor ← n3
    Fin Si
Fin Si

inter ← (n1+n2+n3)-(mayor+menor)

//Salida
Escribir mayor, inter, menor

Fin

```

Codificación:

```

<?php
//Variables
$n1 = 0; $n2 = 0; $n3 = 0;
$mayor = 0; $inter = 0; $menor = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n1 = (int)$_POST["txtn1"];
    $n2 = (int)$_POST["txtn2"];
    $n3 = (int)$_POST["txtn3"];

    //Proceso
    if($n1 > $n2 && $n1 > $n3){
        $mayor = $n1;
    }else{
        if($n2 > $n1 && $n2 > $n3){
            $mayor = $n2;
        }else{
            $mayor = $n3;
        }
    }

    if($n1 < $n2 && $n1 < $n3){
        $menor = $n1;
    }else{
        if($n2 < $n1 && $n2 < $n3){
            $menor = $n2;
        }else{
            $menor = $n3;
        }
    }

    $inter = ($n1 + $n2 + $n3) - ($mayor + $menor);
}
?>
<html>
<head>
<title>Problema 18</title>
<style type="text/css">
<!--

```

```

.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema18.php">
    <table width="258" border="0">
        <tr>
            <td colspan="2"><strong>Problema 18</strong> </td>
        </tr>
        <tr>
            <td width="75">Numero 1 </td>
            <td width="151">
                <input name="txtn1" type="text" id="txtn1" value="<?=$n1?>" />
            </td>
        </tr>
        <tr>
            <td>Numero 2 </td>
            <td><input name="txtn2" type="text" id="txtn2" value="<?=$n2?>"></td>
        </tr>
        <tr>
            <td>Numero 3 </td>
            <td><input name="txtn3" type="text" id="txtn3" value="<?=$n3?>"></td>
        </tr>
        <tr>
            <td>Mayor</td>
            <td><input name="txtmayor" type="text" class="TextoFondo"
id="txtmayor" value="<?=$mayor?>"></td>
        </tr>
        <tr>
            <td>Intermedio</td>
            <td><input name="txtinter" type="text" class="TextoFondo"
id="txtinter" value="<?=$inter?>"></td>
        </tr>
        <tr>
            <td>Menor</td>
            <td><input name="txtmenor" type="text" class="TextoFondo"
id="txtmenor" value="<?=$menor?>"></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 19

Enunciado: Un restaurante ofrece un descuento del 10 % para consumos de hasta S/.100.00; y un descuento del 20 % para consumos mayores. En ambos casos se aplica un impuesto del 19 %. Determinar el monto del descuento, el impuesto y el importe a pagar.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese el consumo; luego, que el sistema verifique y calcule el monto del descuento, el impuesto y el importe a pagar.

Entrada

- Consumo (c)

Salida

- Monto del descuento (m_d)
- Impuesto (m_igv)
- Importe a pagar (p)

**Diseño:
Interfaz de usuario**

Problema 19 - Windows Internet Explorer

http://localhost/Cap03/Problema19.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 19

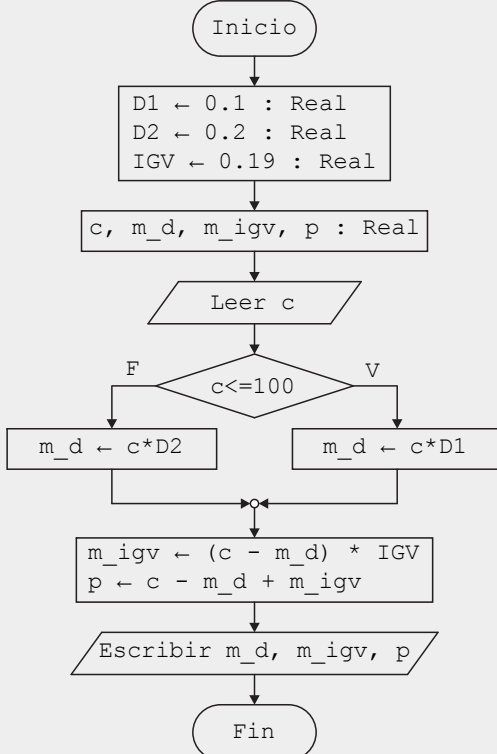
Problema 19

Consumo

Monto descuento

Impuesto IGV

Importe a pagar

Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio****//Constantes**

D1 = 0.1 : Real
D2 = 0.2 : Real
IGV = 0.19 : Real

//Variables

c, m_d, m_igv, p : Real

//Entrada

Leer c

//Proceso

Si c ≤ 100 Entonces

 m_d ← c * D1

SiNo

 m_d ← c * D2

Fin Si

m_igv ← (c - m_d) * IGV

p ← c - m_d + m_igv

//Salida

Escribir m_d, m_igv, p

Fin

Codificación:

```

<?php
//Constantes
define(D1,0.1);
define(D2,0.2);
define(IGV,0.19);

//Variables
$c = 0; $m_d = 0; $m_igv = 0; $p = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $c = (float)$_POST["txtc"];

    //Proceso
    if($c <= 100 ){
        $m_d = $c * D1;
    }else{
        $m_d = $c * D2;
    }

    $m_igv = ($c - $m_d) * IGV;
    $p = $c - $m_d + $m_igv;
}
?>
<html>
<head>
<title>Problema 19</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema19.php">
    <table width="273" border="0">
        <tr>
            <td colspan="2"><strong>Problema 19</strong> </td>
        </tr>
        <tr>
            <td width="75">Consumo</td>
            <td width="151">
                <input name="txtc" type="text" id="txtc" value="<?=$c?>" />
            </td>
        </tr>
        <tr>
            <td>Monto descuento </td>

```

```

        <td><input name="txtm_d" type="text" class="TextoFondo" id="txtm_d"
value="<?=$m_d?>"></td>
    </tr>
    <tr>
        <td>Impuesto IGv </td>
        <td><input name="txtm_igv" type="text" class="TextoFondo"
id="txtm_igv" value="<?=$m_igv?>"></td>
    </tr>
    <tr>
        <td>Importe a pagar </td>
        <td><input name="txtp" type="text" class="TextoFondo" id="txtp"
value="<?=$p?>"></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 20

Enunciado: Debido a los excelentes resultado, el restaurante decide ampliar sus ofertas de acuerdo a la siguiente escala de consumo (ver tabla). Determinar el monto del descuento, el importe del impuesto y el importe a pagar.

Consumo (S/.)	Descuento (%)
Hasta 100	10
Mayor a 100	20
Mayor a 200	30

Análisis: Para la solución de este problema se requiere que el usuario ingrese el consumo; luego, el sistema verifique y calcule el monto del descuento, el impuesto y el importe a pagar.

Entrada

- Consumo (c)

Salida

- Monto del descuento (m_d)
- Impuesto (m_{igv})
- Importe a pagar (p)

**Diseño:
Interfaz de usuario**

Problema 20 - Windows Internet Explorer

http://localhost/Cap03/Problema20.php

Archivo Edición Ver Favoritos Herramientas Ayu

Problema 20

Problema 20

Consumo

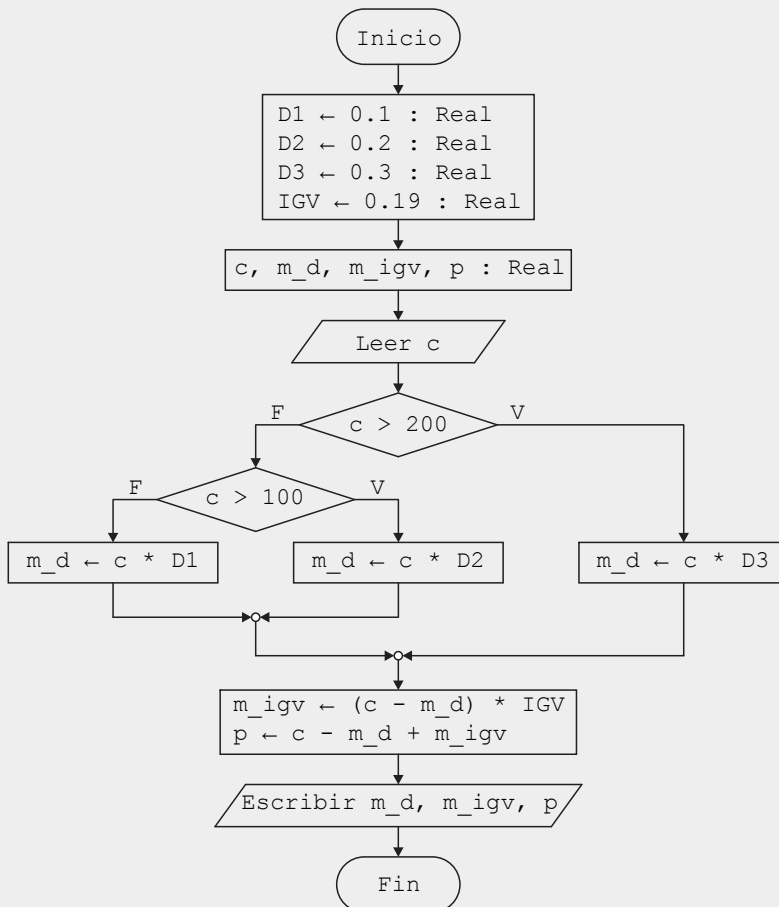
Monto descuento

Impuesto IGV

Importe a pagar

Algoritmo

Diagrama de flujo



Pseudocódigo**Inicio****//Constantes**

D1 = 0.1 : Real

D2 = 0.2 : Real

D3 = 0.3 : Real

IGV = 0.19 : Real

//Variables

c, m_d, m_igv, p : Real

//Entrada

Leer c

//Proceso

Si c > 200 Entonces

m_d ← c * D3

SiNo

Si c > 100 Entonces

m_d ← c * D2

SiNo

m_d ← c * D1

Fin Si

Fin Si

m_igv ← (c - m_d) * IGV

p ← c - m_d + m_igv

//Salida

Escribir m_d, m_igv, p

Fin**Codificación:**

```

<?php
//Constantes
define(D1,0.1);
define(D2,0.2);
define(D3,0.3);
define(IGV,0.19);

//Variables
$c = 0; $m_d = 0; $m_igv = 0; $p = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $c = (float)$_POST["txtc"];

    //Proceso

```

```

    if($c > 200){
        $m_d = $c * D3;
    }else{
        if($c > 100 ){
            $m_d = $c * D2;
        }else{
            $m_d = $c * D1;
        }
    }
    $m_igv = ($c - $m_d) * IGV;
    $p = $c - $m_d + $m_igv;
}
?>
<html>
<head>
<title>Problema 20</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema20.php">
    <table width="273" border="0">
        <tr>
            <td colspan="2"><strong>Problema 20</strong> </td>
        </tr>
        <tr>
            <td width="75">Consumo</td>
            <td width="151">
                <input name="txtc" type="text" id="txtc" value="<?=$c?>" />
            </td>
        </tr>
        <tr>
            <td>Monto descuento </td>
            <td><input name="txtm_d" type="text" class="TextoFondo" id="txtm_d"
value="<?=$m_d?>"></td>
        </tr>
        <tr>
            <td>Impuesto IGV </td>
            <td><input name="txtm_igv" type="text" class="TextoFondo"
id="txtm_igv" value="<?=$m_igv?>"></td>
        </tr>
        <tr>
            <td>Importe a pagar </td>
            <td><input name="txtp" type="text" class="TextoFondo" id="txtp"
value="<?=$p?>"></td>

```

```

    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
            value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 21

Enunciado: Al ingresar el valor de una temperatura, obtener el tipo de clima según la siguiente tabla.

Temperatura	Tipo de clima
Temp. < 10	Frío
Temp. entre 10 Y 20	Nublado
Temp. entre 21 Y 30	Calor
Temp. > 30	Tropical

Análisis: Para la solución de este problema se requiere que el usuario ingrese la temperatura; luego, que el sistema verifique y determine el clima.

Entrada

- Temperatura (t)

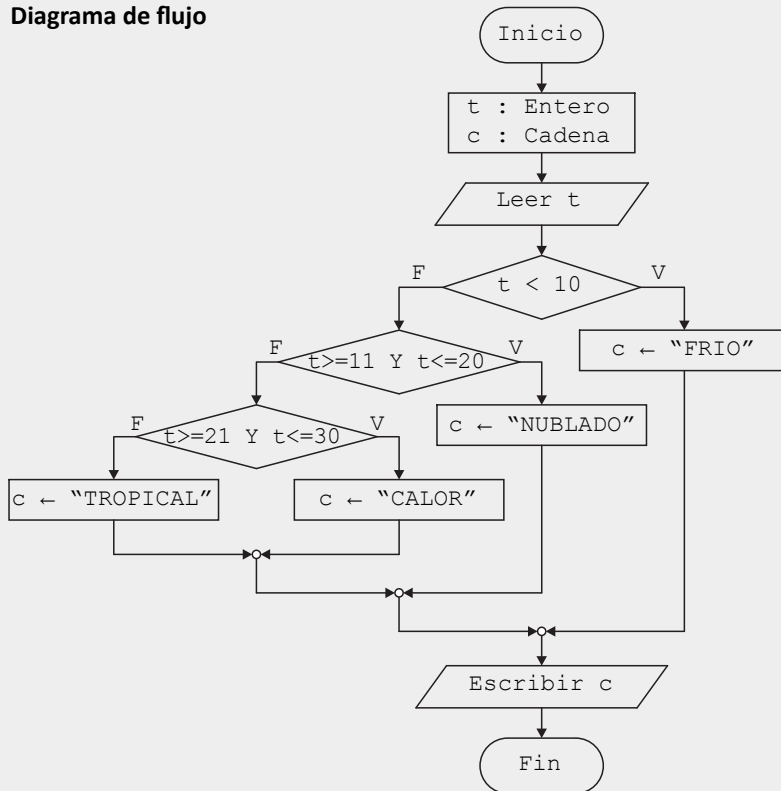
Salida

- Clima (c)

Diseño:

Interfaz de usuario

The screenshot shows a web browser window titled "Problema 21 - Windows Internet Explorer". The address bar displays the URL "http://localhost/Cap03/Problema21.php". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The page content features a form titled "Problema 21" with two input fields: "Temperatura" containing the value "28" and "Clima" containing the value "CALOR". Below these fields is a button labeled "Calcular".

Algoritmo**Diagrama de flujo****Pseudocódigo**

Inicio

//Variablest : Entero
c : Cadena**//Entrada**

Leer t

//ProcesoSi t < 10 Entonces
 c ← "FRIO"

SiNo

 Si t >= 11 Y t <= 20 Entonces
 c ← "NUBLADO"

SiNo

 Si t >= 21 Y t <= 30 Entonces
 c ← "CALOR"

SiNo

c ← "TROPICAL"

Fin Si

Fin Si

Fin Si

//Salida

Escribir c

Fin

Codificación:

```

<?php
//Variables
$t = 0; $c = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $t = (int)$_POST["txtt"];

    //Proceso
    if($t < 10) {
        $c = "FRIO";
    }else{
        if($t >= 10 && $t <= 20){
            $c = "NUBLADO";
        }else{
            if($t >= 21 && $t <= 30){
                $c = "CALOR";
            }else{
                $c = "TROPICAL";
            }
        }
    }
}
?>
<html>
<head>
<title>Problema 21</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema21.php">
    <table width="250" border="0">
        <tr>
            <td colspan="2"><strong>Problema 21</strong> </td>
        </tr>
        <tr>
            <td width="89">Temperatura</td>
            <td width="151">
                <input name="txtt" type="text" id="txtt" value="<?=$t?>" />
            </td>
        </tr>
        <tr>
            <td>Clima</td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>"></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 22

Enunciado: Un negocio tiene dos tipos de cliente: cliente general (G) y cliente afiliado (A). También acepta dos formas de pago: al Contado (C) y en plazos (P). Nos piden crear un programa que al ingresar el monto de la compra se obtenga el **monto del descuento** o el **monto del recargo** y el **total a pagar**, según la siguiente tabla.

Análisis: Para la solución de este problema se requiere que el usuario ingrese el monto de la compra, el tipo de cliente y la forma de pago; luego, que el sistema verifique y determine el monto de descuento o recargo y el total a pagar.

Tipo	Contado (C) Descuento	Plazos (P) Recargo
Cliente general (G)	15 %	10 %
Cliente afiliado (A)	20 %	5 %

Entrada

- Monto de la compra (mc)
- Tipo de cliente (tc)
- Forma de pago (fp)

Salida

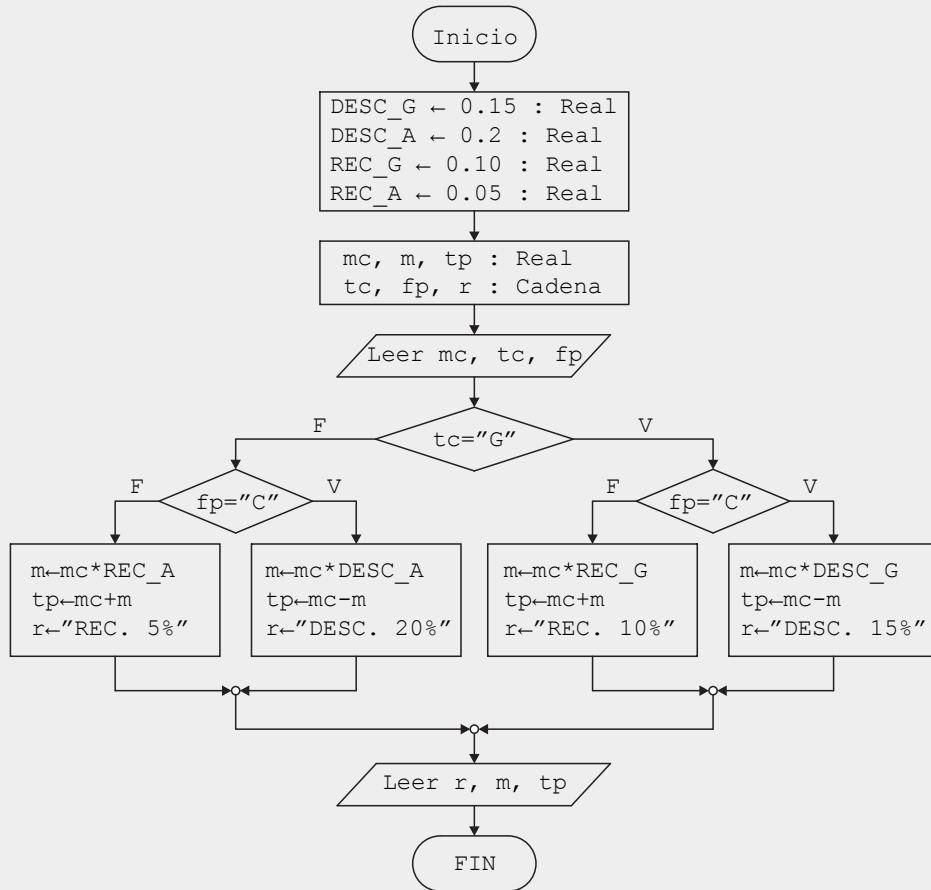
- Monto de descuento o recargo (m)
- Total a pagar (tp)

Diseño:**Interfaz de usuario**

The screenshot shows a web browser window titled "Problema 21 - Windows Internet Explorer". The address bar shows "http://localhost/Cap03/Problema22.php". The browser has a menu bar with "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". Below the menu bar is a search bar and a "Problema 21" icon. The main content area displays a form titled "Problema 21" with the following fields and values:

Monto de compra	100
Tipo de cliente	A
Forma de pago	C
RECARGO 5%	5
Total a pagar	105

At the bottom right of the form is a "Calcular" button.

Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio****//Constantes**

```

DESC_G = 0.15 : Real
DESC_A = 0.2 : Real
REC_G = 0.10 : Real
REC_A = 0.05 : Real

```

//Variables

```

mc, m, tp : Real
tc, fp, r : Cadena

```

//Entrada

```

Leer mc, tc, fp

```

//Proceso

```

Si tc = "G" Entonces
    Si fp = "C" Entonces
        m ← mc * DESC_G
        tp ← mc - m

```

```

        r ← "DESCUENTO 15%"
    SiNo
        m ← mc * REC_G
        tp ← mc + m
        r ← "RECARGA 10%"
    Fin Si
SiNo
    Si fp = "C" Entonces
        m ← mc * DESC_A
        tp ← mc - m
        r ← "DESCUENTO 20%"

    SiNo
        m ← mc * REC_A
        tp ← mc + m
        r ← "RECARGA 5%"
    Fin Si
Fin Si

//Salida
Escribir r, m, tp

```

Fin

Codificación:

```

<?php
//Constantes
define(DESC_G, 0.15);
define(DESC_A, 0.2);
define(REC_G, 0.1);
define(REC_A, 0.05);

//Variables
$mc = 0; $m = 0; $tp = 0;
$tc = ""; $fp = ""; $r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $mc = (float)$_POST["txtmc"];
    $tc = $_POST["txttc"];
    $fp = $_POST["txtfp"];

    //Proceso
    if($tc == 'G'){
        if($fp == 'C'){
            $m = $mc * DESC_G;
            $tp = $mc - $m;
            $r = "DESCUENTO 15%";
        }else{
            $m = $mc * REC_G;
            $tp = $mc + $m;
            $r = "RECARGO 10%";
        }
    }else{
        if(fp == 'C'){
            $m = $mc * DESC_A;
            $tp = $mc - $m;
            $r = "DESCUENTO 20%";
        }else{

```

```

        $m = $mc * REC_A;
        $tp = $mc + $m;
        $r = "RECARGO 5%";
    }
}
?>
<html>
<head>
<title>Problema 22</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema22.php">
    <table width="277" border="0">
        <tr>
            <td colspan="2"><strong>Problema 21</strong> </td>
        </tr>
        <tr>
            <td width="116">Monto de compra </td>
            <td width="151">
                <input name="txtmc" type="text" id="txtmc" value="<?=$mc?>" />
            </td>
        </tr>
        <tr>
            <td>Tipo de cliente </td>
            <td><input name="txttc" type="text" id="txttc" value="<?=$tc?>"></td>
        </tr>
        <tr>
            <td>Forma de pago </td>
            <td><input name="txtfp" type="text" id="txtfp" value="<?=$fp?>"></td>
        </tr>
        <tr>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>"></td>
            <td><input name="txtm" type="text" class="TextoFondo" id="txtm"
value="<?=$m?>"></td>
        </tr>
        <tr>
            <td>Total a pagar </td>
            <td><input name="txttp" type="text" class="TextoFondo" id="txttp"
value="<?=$tp?>"></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 23

Enunciado: Elabore un algoritmo que resuelva una ecuación de primer grado.

$$ax + b = 0 \quad x = \frac{-b}{a}$$

Considerar si **a** es diferente a 0 no es una ecuación de primer grado.

Análisis: Para la solución de este problema se requiere que el usuario ingrese el valor de a y b; luego, que el sistema verifique y determine el valor de x.

Entrada

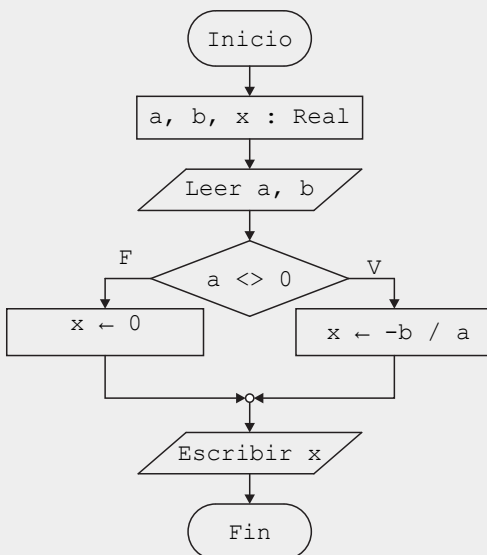
- Coeficiente a (a)
- Término independiente b (b)

Salida

- Raíz x (x)

Diseño:**Interfaz de usuario**

The screenshot shows a web browser window titled 'Problema 23 - Windows Internet Explorer'. The address bar shows 'http://localhost/Cap03/Problema23.php'. The browser menu includes 'Archivo', 'Edición', 'Ver', 'Favoritos', 'Herramientas', and 'Ayuda'. The page content is titled 'Problema 23' and contains three input fields: 'a' with the value '-5', 'b' with the value '15', and 'x' with the value '3'. Below the input fields is a button labeled 'Calcular'.

Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio**

//Variables

a, b, x : Real

//Entrada

Leer a, b

//Proceso

Si a <> 0 Entonces

 x ← -b / a

SiNo

 x ← 0

Fin Si

//Salida

Escribir x

Fin

Codificación:

```

<?php

//Variables
$a = 0; $b = 0; $x= 0;

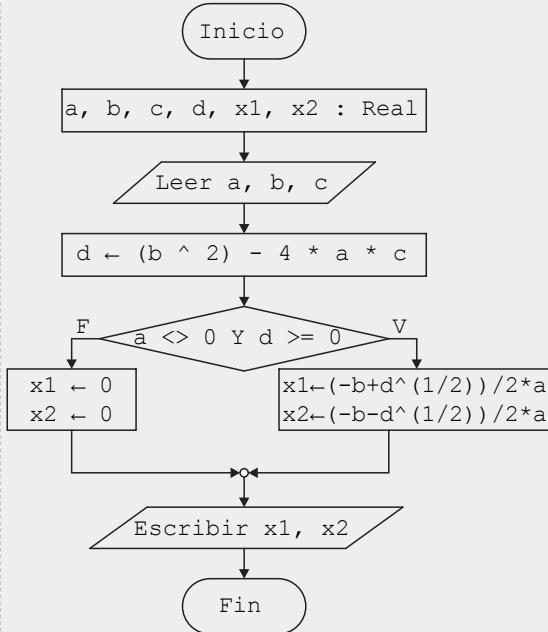
if(isset($_POST["btnCalcular"])) {
    //Entrada
    $a = (float)$_POST["txta"];
    $b = (float)$_POST["txtb"];

    //Proceso
    if($a != 0){
        $x = -$b / $a;
    }else{
        $x = 0;
    }
}
?>
<html>
<head>
<title>Problema 23</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema23.php">
    <table width="241" border="0">
        <tr>
            <td colspan="2"><strong>Problema 23</strong> </td>
        </tr>
        <tr>
            <td width="81">a</td>
            <td width="150">
                <input name="txta" type="text" id="txta" value="<?=$a?>" />
            </td>
        </tr>
        <tr>
            <td>b </td>
            <td>
                <input name="txtb" type="text" id="txtb" value="<?=$b?>" />
            </td>
        </tr>
        <tr>
            <td>x</td>

```


Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

a, b, c, x1, x2 : Real

//Entrada

Leer a, b, c

//Proceso

$d \leftarrow (b^2) - 4 * a * c$

Si $a \neq 0$ Y $d \geq 0$ Entonces

$x1 \leftarrow (-b + d^{(1/2)}) / 2 * a$

$x2 \leftarrow (-b - d^{(1/2)}) / 2 * a$

SiNo

$x1 \leftarrow 0$

$x2 \leftarrow 0$

Fin Si

//Salida

Escribir x1, x2

Fin

Codificación:

```

<?php

//Variables
$a = 0; $b = 0; $c = 0; $x1 = 0; $x2 = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $a = (float)$_POST["txta"];
    $b = (float)$_POST["txtb"];
    $c = (float)$_POST["txtc"];

    //Proceso
    $d = pow($b, 2.0) - 4.0 * $a * $c;
    if($a != 0 && $d >= 0){
        $x1 = (-$b + pow($d, (1.0 / 2.0))) / 2 * $a;
        $x2 = (-$b - pow($d, (1.0 / 2.0))) / 2 * $a;
    }else{
        $x1 = 0;
        $x2 = 0;
    }
}
?>
<html>
<head>
<title>Problema 24</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
  
```

```

-->
</style>
</head>
<body>
<form method="post" action="Problema24.php">
  <table width="241" border="0">
    <tr>
      <td colspan="2"><strong>Problema 24</strong> </td>
    </tr>
    <tr>
      <td width="81">a</td>
      <td width="150">
        <input name="txta" type="text" id="txta" value="<?=$a?" />
      </td>
    </tr>
    <tr>
      <td>b </td>
      <td>
        <input name="txtb" type="text" id="txtb" value="<?=$b?" />
      </td>
    </tr>
    <tr>
      <td>c</td>
      <td>
        <input name="txtc" type="text" id="txtc" value="<?=$c?" />
      </td>
    </tr>
    <tr>
      <td>x1</td>
      <td><input name="txtx1" type="text" class="TextoFondo" id="txtx1"
value="<?=$x1?" /></td>
    </tr>
    <tr>
      <td>x2</td>
      <td><input name="txtx2" type="text" class="TextoFondo" id="txtx2"
value="<?=$x2?" /></td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td>
        <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
      </td>
    </tr>
  </table>
</form>
</body>
</html>

```

Problema n.º 25

Enunciado: Dada la hora, minuto y segundo, encuentre la hora del siguiente segundo.

Análisis: Para la solución de este problema se requiere que el usuario ingrese la hora, minuto y segundo; luego que el sistema verifique y determine la hora, minuto y segundo del siguiente segundo.

Entrada

- Hora (h)
- Minuto (m)
- Segundo (s)

Salida

- Hora (h)
- Minuto (m)
- Segundo (s)

Diseño: Interfaz de usuario

Problema 25 - Windows Internet Explorer

http://localhost/Cap03/Problema25.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 25

Problema 25

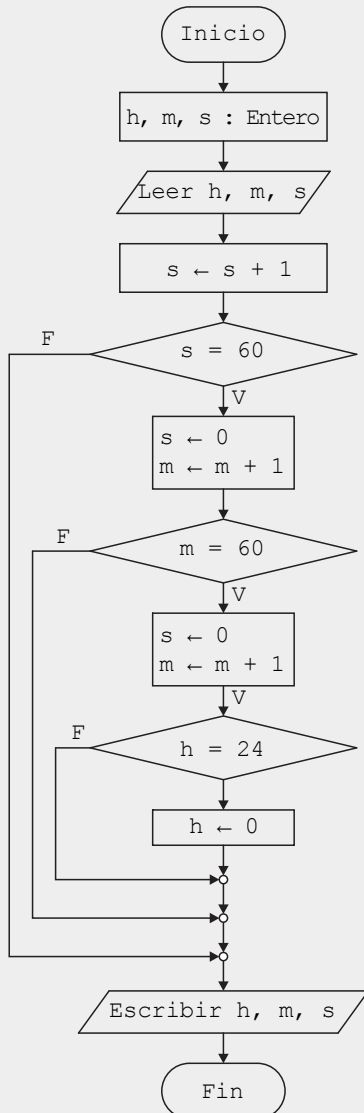
Hora

Minuto

Segundo

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

h, m, s : Entero

//Entrada

Leer h, m, s

//Proceso

s ← s + 1

Si s = 60 Entonces

s ← 0

m ← m + 1

Si m = 60 Entonces

m ← 0

h ← h + 1

Si h = 60 Entonces

h ← 0

Fin Si

Fin Si

Fin Si

//Salida

Escribir h, m, s

Fin

Codificación:

```

<?php

//Variables
$h = 0; $m = 0; $s = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $h = (float)$_POST["txth"];
    $m = (float)$_POST["txtm"];
    $s = (float)$_POST["txts"];

    //Proceso
    $s += 1;
    if($s == 60){
        $s = 0;
        $m += 1;
        if($m == 60){
            $m = 0;
            $h += 1;
            if($h == 24){
                $h = 0;
            }
        }
    }
}

?>
<html>
<head>
<title>Problema 25</title>
</head>
<body>
<form method="post" action="Problema25.php">
    <table width="241" border="0">
        <tr>
            <td colspan="2"><strong>Problema 25</strong> </td>
        </tr>
        <tr>
            <td width="81">Hora</td>
            <td width="150">
                <input name="txth" type="text" id="txth" value="<?=$h?>" />
            </td>
        </tr>
        <tr>
            <td>Minuto </td>
            <td>
                <input name="txtm" type="text" id="txtm" value="<?=$m?>" />
            </td>
        </tr>
    </table>

```

```
<tr>
  <td>Segundo</td>
  <td>
    <input name="txts" type="text" id="txts" value="<?=$s?>"/>
  </td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>
    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
  </td>
</tr>
</table>
</form>
</body>
</html>
```

3.5 Problemas propuestos

Los siguientes problemas le servirán para medir su aprendizaje, es importante que los resuelva.

Propuesto n.º 11

Enunciado: Dada la edad de una persona, determinar si es mayor o menor de edad; considere que son mayores de edad aquellas personas que tienen de 18 años a más.

Propuesto n.º 12

Enunciado: Si tenemos dos números enteros, devolver el número menor.

Propuesto n.º 13

Enunciado: Si tenemos dos números, determinar si son iguales o son diferentes.

Propuesto n.º 14

Enunciado: Dado un número entero, devolver el doble del número si este es positivo; el triple, si es negativo; y cero si el número es neutro.

Propuesto n.º 15

Enunciado: Crear un programa que al ingresar tres números enteros, los devuelva ordenados en forma ascendente y en forma descendente.

Propuesto n.º 16

Enunciado: Después de ingresar 4 notas, obtener el promedio de las tres mejores y mostrar el mensaje «Aprobado», si el promedio es mayor o igual a 11; caso contrario, mostrar «Desaprobado».

Propuesto n.º 17

Enunciado: Dados los siguientes datos de entrada: «saldo anterior», tipo de movimiento «R» (retiro) o «D» (depósito) y «monto de la transacción», obtener como dato de salida el saldo actual.

Propuesto n.º 18

Enunciado: Dados 2 números enteros a y b , determinar cuál es mayor con respecto al otro.

a es mayor que **b**

b es mayor que **a**

a es igual a **b**

Propuesto n.º 19

Enunciado: Dado que se tienen tres longitudes, diga si forman un triángulo.

Teorema: En todo triángulo, cada lado es menor que la suma de los otros dos, pero mayor que su diferencia.

Propuesto n.º 20

Enunciado: Dado que tenemos tres longitudes; si forman un triángulo, devolver el tipo de triángulo según sus lados.

T. equilátero: Sus 3 lados son iguales.

T. isósceles: 2 lados iguales.

T. escaleno: 3 lados diferentes.

Estructura selectiva múltiple

4.1 Introducción

Sabemos que en la actualidad existen muchos sistemas financieros que ofrecen préstamos con condiciones diferentes; usted, al solicitar un préstamo, tiene que evaluar diversas alternativas y decidirse por una de ellas.

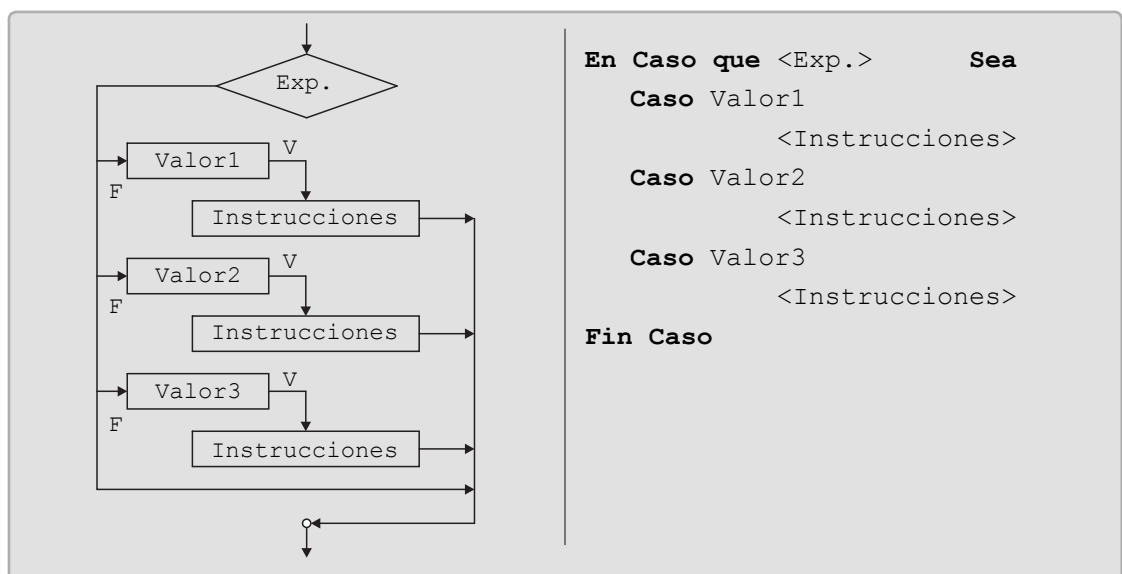
En los lenguajes de programación se cuenta con una implementación similar, la cual recibe el nombre de estructura selectiva múltiple, que permite evaluar varias alternativas y realizar el proceso para comprobar si cumple o no con la condición elegida.

Muchas veces, para solucionar este tipo de problemas, se utilizan estructuras selectivas dobles anidadas (en cascada), dando una solución muy complicada y confusa para analizar; es recomendable que cuando se tenga que evaluar varias alternativas se utilice estructuras selectiva múltiple porque es más legible, eficiente y fácil de interpretar.

4.2 Estructura selectiva múltiple

Permite comparar un valor con diversas alternativas; si la comparación tiene éxito, se ejecuta el grupo de instrucción que contenga la alternativa seleccionada y, luego, sale de la estructura.

Muchas se pueden implementar, en forma opcional, una alternativa por defecto; es decir, si al comparar con todas las alternativas propuestas no se tiene éxito con ninguna, entonces se ejecuta la alternativa por defecto.

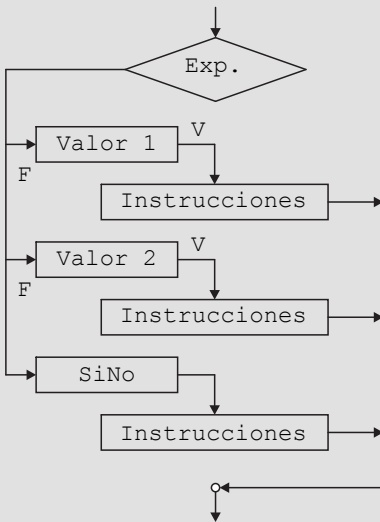


Sintaxis 1 PHP

```

switch (<Exp.>) {
    case Valor1:
        <Instrucciones>;
        break;
    case Valor2:
        <Instrucciones>;
        break;
    case Valor3:
        <Instrucciones>;
        break;
}

```



En Caso que <Exp.> Sea
Caso Valor1
 <Instrucciones>
Caso Valor2
 <Instrucciones>
SiNo
 <Instrucciones>
Fin Caso

Sintaxis 2 PHP

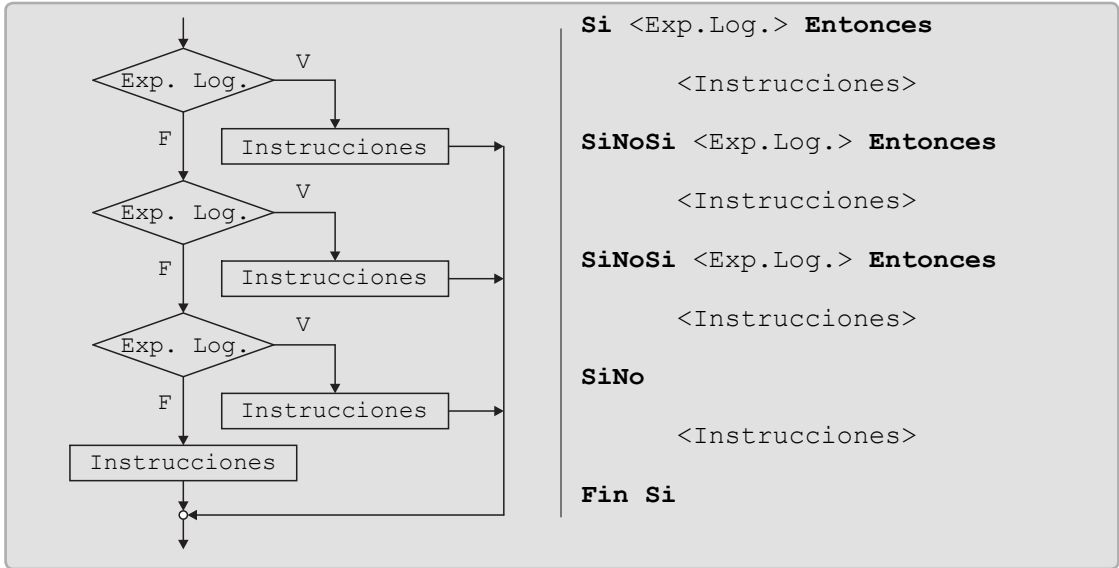
```

switch (<Exp.>) {
    case Valor1:
        <Instrucciones>;
        break;
    case Valor2:
        <Instrucciones>;
        break;
    default:
        <Instrucciones>;
        break;
}

```

4.2.1 Estructura selectiva múltiple usando rangos

La estructura selectiva múltiple permite comparar un valor (igualdad), pero cuando se requiere manejar rangos ($\geq$ Y $\leq$) se puede usar una estructura selectiva múltiple similar a la estructura selectiva doble anidada.



Sintaxis PHP

```

if (<Exp. Log.>)
    <Instrucciones>;
elseif (<Exp. Log.>)
    <Instrucciones>;
elseif (<Exp. Log.>)
    <Instrucciones>;
else
    <Instrucciones>;
  
```

Problema n.º 26

Enunciado: Al ingresar un número entre 1 y 4, devolver la estación del año de acuerdo a la siguiente tabla.

Número	Estación
1	Verano
2	Otoño
3	Invierno
4	Primavera

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número entero; luego, que el sistema realice el proceso para devolver la estación.

Entrada

- Número (n)

Salida

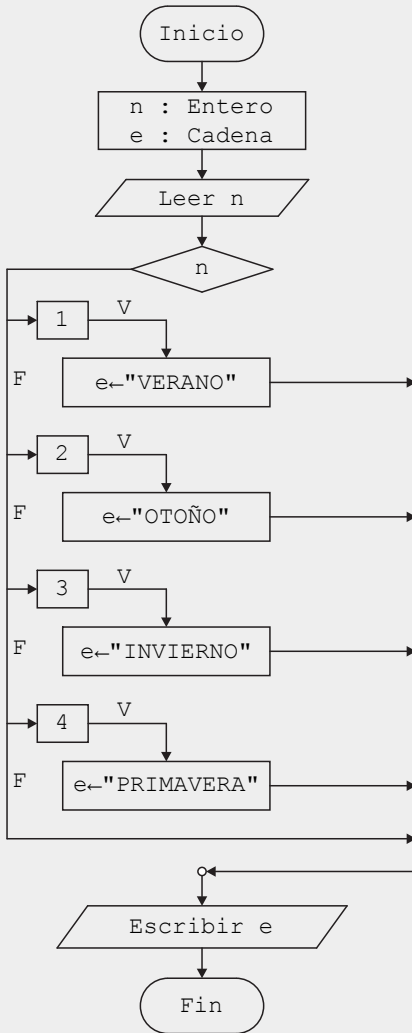
- Estación (e)

Diseño:**Interfaz de usuario**

The screenshot shows a web browser window titled "Problema 26 - Windows Internet Explorer". The address bar displays "http://localhost/Cap04/Problema26.php". The browser menu includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayu...". The page content is titled "Problema 26" and contains two input fields: "Numero" with the value "3" and "Estacion" with the value "INVIERNO". Below these fields is a "Calcular" button.

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

n : Entero

e : Cadena

//Entrada

Leer n

//Proceso

En Caso que n Sea

Caso 1

e ← "VERANO"

Caso 2

e ← "INVIERNO"

Caso 3

e ← "OTOÑO"

Caso 4

e ← "PRIMAVERA"

Fin Caso

//Salida

Escribir e

Fin

Codificación:

```
<?php

//Variables
$n = 0; $e = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (float)$_POST["txtn"];

    //Proceso
    switch($n) {
        case 1:
            $e = "VERANO";
            break;
        case 2:
            $e = "OTOÑO";
            break;
        case 3:
            $e = "INVIERNO";
            break;
        case 4:
            $e = "PRIMAVERA";
            break;
    }
}
?>
<html>
<head>
<title>Problema 26</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema26.php">
    <table width="241" border="0">
        <tr>
            <td colspan="2"><strong>Problema 26</strong> </td>
        </tr>
        <tr>
            <td width="81">Numero</td>
            <td width="150">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>
```

```

<tr>
  <td>Estacion </td>
  <td>
    <input name="txte" type="text" class="TextoFondo" id="txte"
value="<?=$e?>" />
  </td>
</tr>

<tr>
  <td>&nbsp;</td>
  <td>
    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
  </td>
</tr>
</table>
</form>
</body>
</html>

```

Problema n.º 27

Enunciado: Dado un número entero de un dígito (0 al 9), devolver el número en letras.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número entero; luego, que el sistema verifique y devuelva el número en letras.

Entrada

- Número (n)

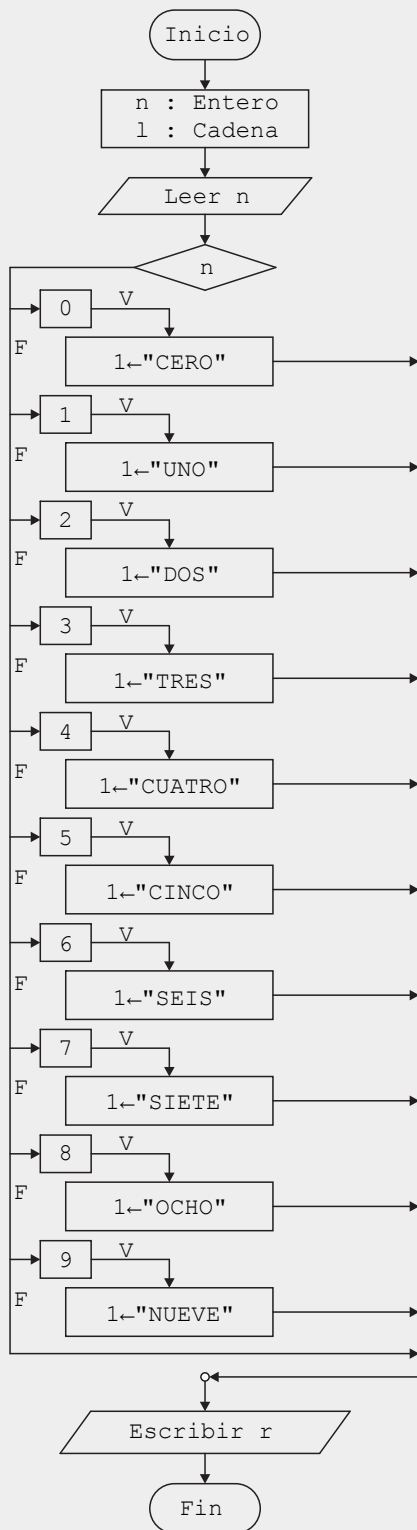
Salida

- Resultado (r)

Diseño:

Interfaz de usuario

The screenshot shows a web browser window with the title "Problema 27 - Windows Internet Explorer". The address bar displays the URL "http://localhost/Cap04/Problema27.php". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". Below the menu bar, there is a star icon and the text "Problema 27". The main content area of the browser displays a form titled "Problema 27". The form contains two input fields: "Numero" with the value "9" and "Letra" with the value "NUEVE". Below these fields is a button labeled "Calcular".

Diagrama de flujo**Algoritmo****Inicio****//Variables**

n : Entero

l : Cadena

//Entrada

Leer n

//Proceso

En Caso que n Sea

Caso 0

l ← "CERO"

Caso 1

l ← "UNO"

Caso 2

l ← "DOS"

Caso 3

l ← "TRES"

Caso 4

l ← "CUATRO"

Caso 5

l ← "CINCO"

Caso 6

l ← "SEIS"

Caso 7

l ← "SIETE"

Caso 8

l ← "OCHO"

Caso 7

l ← "NUEVE"

Fin Caso

//Salida

Escribir l

Fin

Codificación:

```
<?php

//Variables
$n = 0; $l = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (float)$_POST["txtn"];

    //Proceso
    switch($n) {
        case 0:
            $l = "CERO";
            break;
        case 1:
            $l = "UNO";
            break;
        case 2:
            $l = "DOS";
            break;
        case 3:
            $l = "TRES";
            break;
        case 4:
            $l = "CUATRO";
            break;
        case 5:
            $l = "CINCO";
            break;
        case 6:
            $l = "SEIS";
            break;
        case 7:
            $l = "SIETE";
            break;
        case 8:
            $l = "OCHO";
            break;
        case 9:
            $l = "NUEVE";
            break;
    }
}

?>
<html>
<head>
<title>Problema 27</title>
<style type="text/css">
<!--
```

```

.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema27.php">
    <table width="241" border="0">
        <tr>
            <td colspan="2"><strong>Problema 27</strong> </td>
        </tr>
        <tr>
            <td width="81">Numero</td>
            <td width="150">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Letra</td>
            <td>
                <input name="txtl" type="text" class="TextoFondo" id="txtl"
value="<?=$l?>" />
            </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 28

Enunciado: Dados dos números enteros y un operador (+, -, * y /), devolver la operación de los dos números según el operador ingresado. Considere que si el segundo número es cero y el operador es /, no es divisible con el primer número, entonces devolver como resultado 0.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese un operador y dos números; luego, que el sistema verifique la operación y devuelva el resultado de la operación.

Entrada

- Operador (op)
- Número (n1 y n2)

Salida

- Resultado (r)

Diseño:**Interfaz de usuario**

Problema 28 - Windows Internet Explorer

http://localhost/Cap04/Problema28.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 28

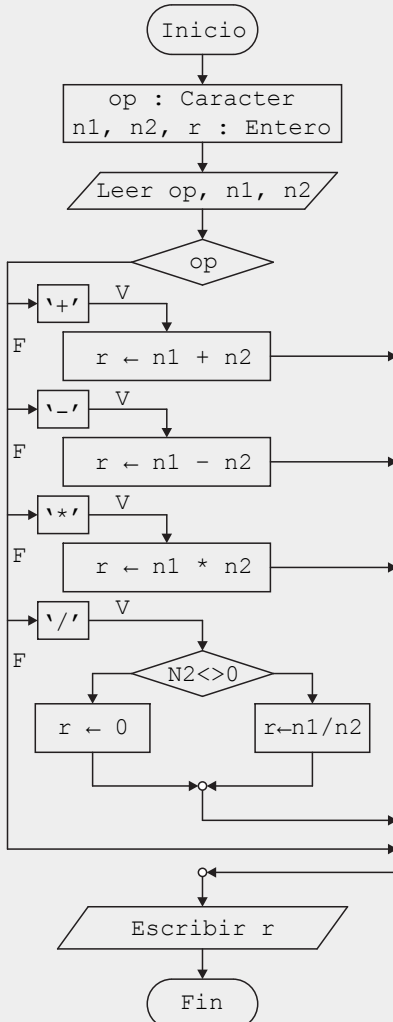
Problema 28

Operacion

Numero 1

Numero 2

Resultado

Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio****//Variables**

op : Caracter
n1, n2, r : Entero

//Entrada

Leer op, n1, n2

//Proceso

En Caso que op Sea

Caso '+'

r ← n1 + n2

Caso '-'

r ← n1 - n2

Caso '*'

r ← n1 * n2

Caso '/'

Si n2 <> 0 Entonces

r ← n1 / n2

SiNo

r ← 0

Fin Sin

Fin Caso

//Salida

Escribir r

Fin

Codificación:

```

<?php

//Variables
$n1 = 0; $n2 = 0; $r = 0; $op = '';

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $op = $_POST["txtop"];
    $n1 = (int)$_POST["txtn1"];
    $n2 = (int)$_POST["txtn2"];

    //Proceso
    if($op == '+') {
        $r = $n1 + $n2;
    }elseif($op == '-') {
        $r = $n1 - $n2;
    }elseif($op == '*') {
        $r = $n1 * $n2;
    }elseif($op == '/') {
        if($n2 != 0)
            $r = $n1 / $n2;
        else
            $r = 0;
    }
}

?>
<html>
<head>
<title>Problema 28</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema28.php">
    <table width="241" border="0">
        <tr>
            <td colspan="2"><strong>Problema 28</strong> </td>
        </tr>
        <tr>
            <td width="81">Operacion</td>
            <td width="150">
                <input name="txtop" type="text" id="txtop" value="<?=$op?>"
            />
            </td>
        </tr>
        <tr>
            <td>Numero 1 </td>
            <td>
                <input name="txtn1" type="text" id="txtn1" value="<?=$n1?>" />
            </td>
        </tr>
        <tr>
            <td>Numero 2 </td>

```

```

        <td><input name="txtn2" type="text" id="txtn2" value="<?=$n2?>" /></td>
    </tr>
    <tr>
        <td>Resultado</td>
        <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 29

Enunciado: Dada una letra, determinar si es una vocal.

Análisis: Para la solución de este problema se requiere que el usuario ingrese una letra «l»; luego, que el sistema analice y determine si es una vocal.

Entrada

- Letra (l)

Salida

- Resultado (r)

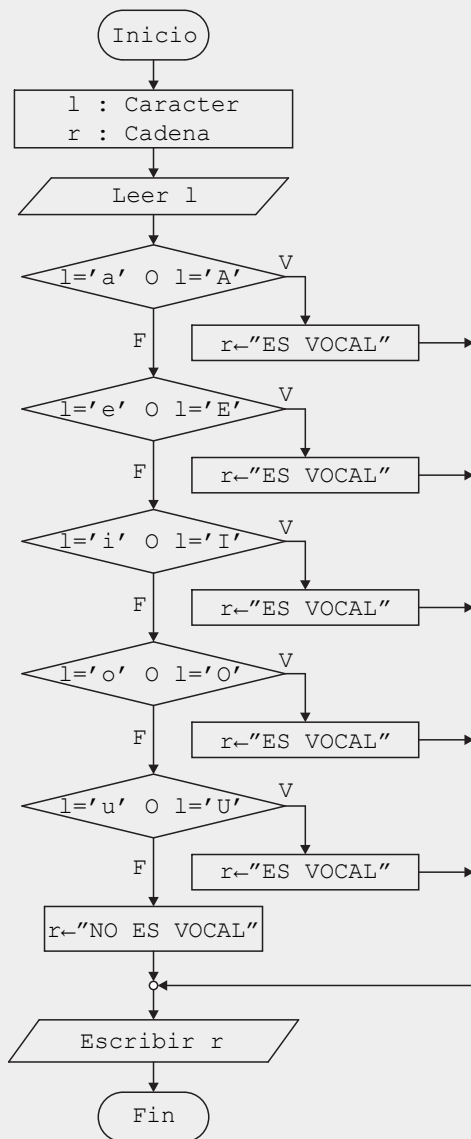
Diseño:

Interfaz de usuario

The screenshot shows a web browser window with the title "Problema 29 - Windows Internet Explorer". The address bar displays the URL "http://localhost/Cap04/Problema29.php". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". Below the menu bar, there are star and plus icons for bookmarks, and a tab labeled "Problema 29". The main content area of the browser shows a web form titled "Problema 29". The form contains two labels: "Letra" and "Resultado". The "Letra" label is next to a text input field that contains the letter "E". The "Resultado" label is next to a text input field that contains the text "ES VOCAL". Below these input fields is a button labeled "Calcular".

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

l : Caracter

r : Cadena

//Entrada

Leer **l**

//Proceso

Si **l** = 'a' O **l** = 'A' Entonces

r ← "ES VOCAL"

SiNoSi **l** = 'e' O **l** = 'E' Entonces

r ← "ES VOCAL"

SiNoSi **l** = 'i' O **l** = 'I' Entonces

r ← "ES VOCAL"

SiNoSi **l** = 'o' O **l** = 'O' Entonces

r ← "ES VOCAL"

SiNoSi **l** = 'u' O **l** = 'U' Entonces

r ← "ES VOCAL"

SiNo

r ← "NO ES VOCAL"

Fin Si

//Salida

Escribir **r**

Fin

Codificación:

```

<?php

//Variables
$l = ''; $r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $l = $_POST["txt1"];

    //Proceso
    if($l == 'a' || $l == 'A')
        $r = "ES VOCAL";
    elseif($l == 'e' || $l == 'E')
        $r = "ES VOCAL";
    elseif($l == 'i' || $l == 'I')
        $r = "ES VOCAL";
    elseif($l == 'o' || $l == 'O')
        $r = "ES VOCAL";
    elseif($l == 'u' || $l == 'U')
        $r = "ES VOCAL";
    else
        $r = "NO ES VOCAL";
}
?>
<html>
<head>
<title>Problema 29</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema29.php">
    <table width="241" border="0">
        <tr>
            <td colspan="2"><strong>Problema 29</strong> </td>
        </tr>
        <tr>
            <td width="81">Letra</td>
            <td width="150">
                <input name="txt1" type="text" id="txt1" value="<?=$l?>" />
            </td>
        </tr>
        <tr>
            <td>Resultado</td>

```

```

        <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 30

Enunciado: Al ingresar el número de un mes, devolver la estación del año de acuerdo a la siguiente tabla.

Mes	Estación
1, 2, 3	Verano
4, 5, 6	Otoño
7, 8, 9	Invierno
10, 11, 12	Primavera

Análisis: Para la solución de este problema se requiere que el usuario ingrese el número del mes; luego, que el sistema verifique y determine la estación.

Entrada

- Mes (m)

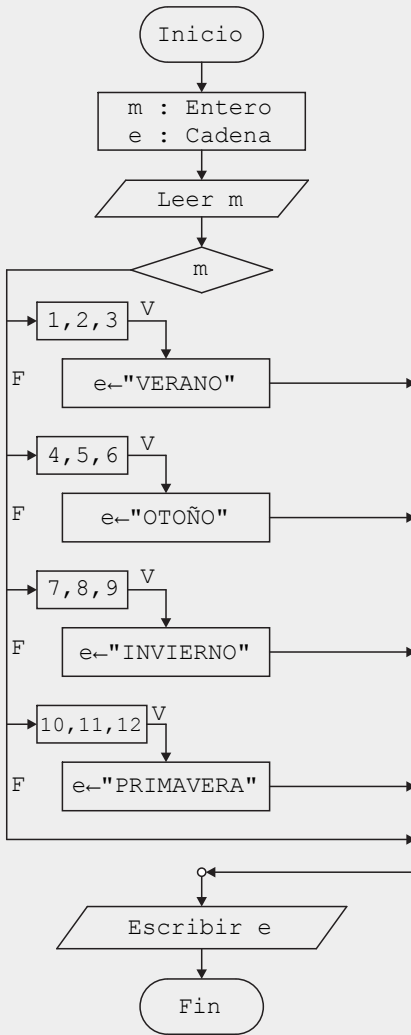
Salida

- Estación (e)

Diseño:

Interfaz de usuario

Diagrama de flujo



Algoritmo

Pseudocódigo

Inicio**//Variables**

m : Entero

e : Cadena

//Entrada

Leer m

//Proceso

En Caso que m Sea

Caso 1,2,3

e ← "VERANO"

Caso 4,5,6

e ← "OTOÑO"

Caso 7,8,9

e ← "INVIERNO"

Caso 10,11,12

e ← "PRIMAVERA"

Fin Caso

//Salida

Escribir e

Fin

Codificación:

```

<?php

//Variables
$m = 0; $e = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $m = $_POST["txtm"];

    //Proceso
    switch($m){
        case 1:
        case 2:
        case 3:
            $e = "VERANO";
            break;
        case 4:
        case 5:
        case 6:
            $e = "OTOÑO";
            break;
        case 7:
        case 8:
        case 9:
            $e = "INVIERNO";
            break;
        case 10:
        case 11:
        case 12:
            $e = "PRIMAVERA";
            break;
    }
}
?>
<html>
<head>
<title>Problema 30</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema30.php">
    <table width="241" border="0">
        <tr>
            <td colspan="2"><strong>Problema 30</strong> </td>
        </tr>
        <tr>
            <td width="81">Mes</td>
            <td width="150">
                <input name="txtm" type="text" id="txtm" value="<?=$m?>" />
            </td>
        </tr>
    </table>
</form>

```

```

</tr>
<tr>
  <td>Estacion</td>
  <td><input name="txte" type="text" class="TextoFondo" id="txte"
value="<?=$e?>" /></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td>
    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
  </td>
</tr>
</table>
</form>
</body>
</html>

```

Problema n.º 31

Enunciado: Dada la nota promedio de un alumno, obtener la categoría, según la siguiente tabla.

Promedio	Categoría
Entre 0 y 5	Pésimo
Entre 6 y 10	Malo
Entre 11 y 14	Regular
Entre 15 y 17	Bueno
Entre 18 y 20	Excelente

Análisis: Para la solución de este problema se requiere que el usuario ingrese el promedio; luego, que el sistema verifique y devuelva la categoría.

Entrada

- Promedio (p)

Salida

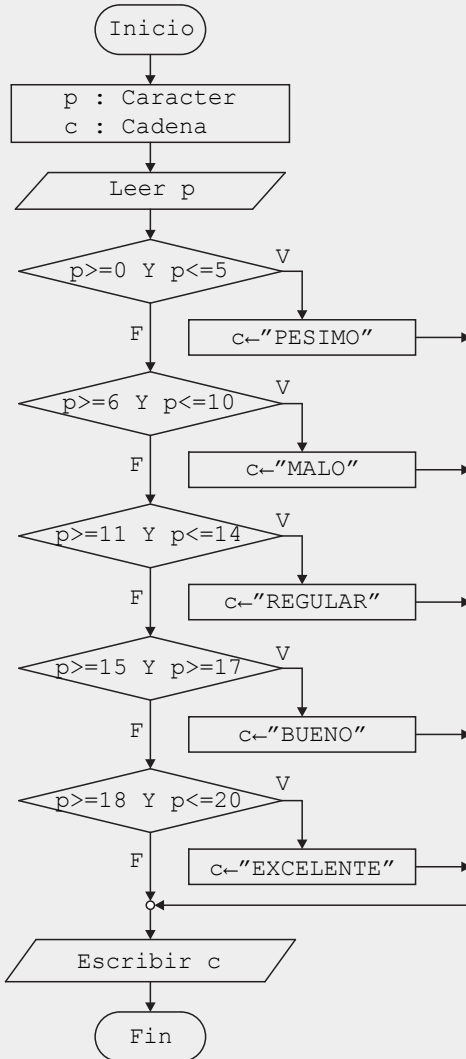
- Categoría (c)

Diseño:

Interfaz de usuario

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

p : Entero

c : Cadena

//Entrada

Leer p

//Proceso

Si p >= 0 Y p <= 5 Entonces

c ← "PESIMO"

SiNo Si p >= 6 Y p <= 10 Entonces

c ← "MALO"

SiNo Si p >= 11 Y p <= 14 Entonces

c ← "REGULAR"

SiNo Si p >= 15 Y p <= 17 Entonces

c ← "BUENO"

SiNo Si p >= 18 Y p <= 20 Entonces

c ← "EXCELENTE"

Fin Si

//Salida

Escribir c

Fin

Codificación:

```

<?php

//Variables
$p = 0; $c = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $p = $_POST["txtp"];

    //Proceso
    if($p >= 0 && $p <= 5){

```

```

        $c = "PESIMO";
    }elseif($p >= 6 && $p <= 10){
        $c = "MALO";
    }elseif($p >= 11 && $p <= 14){
        $c = "REGULAR";
    }elseif($p >= 15 && $p <= 17){
        $c = "BUENO";
    }elseif($p >= 18 && $p <= 20){
        $c = "EXCELENTE";
    }
}
?>
<html>
<head>
<title>Problema 31</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema31.php">
    <table width="241" border="0">
        <tr>
            <td colspan="2"><strong>Problema 31</strong> </td>
        </tr>
        <tr>
            <td width="81">Promedio</td>
            <td width="150">
                <input name="txtp" type="text" id="txtp" value="<?=$p?>" />
            </td>
        </tr>
        <tr>
            <td>Categoria</td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>"/></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 32

Enunciado: Al ingresar el día y el número de un mes, devolver la estación del año de acuerdo a la siguiente tabla.

Estación	Tiempo
Verano	Del 21 de diciembre al 20 de marzo
Otoño	Del 21 de marzo al 21 de junio
Invierno	Del 22 de junio al 22 de septiembre
Primavera	Del 23 de septiembre al 20 de diciembre

Análisis: Para la solución de este problema se requiere que el usuario ingrese el día y el mes; luego, que el sistema verifique y devuelva la estación.

Entrada

- Día (d)
- Mes (m)

Salida

- Estación (e)

Diseño:**Interfaz de usuario**

Problema 32 - Windows Internet Explorer

http://localhost/Cap04/Problema32.php

Archivo Edición Ver Favoritos Herramientas Ayuda

☆ + Problema 32

Problema 32

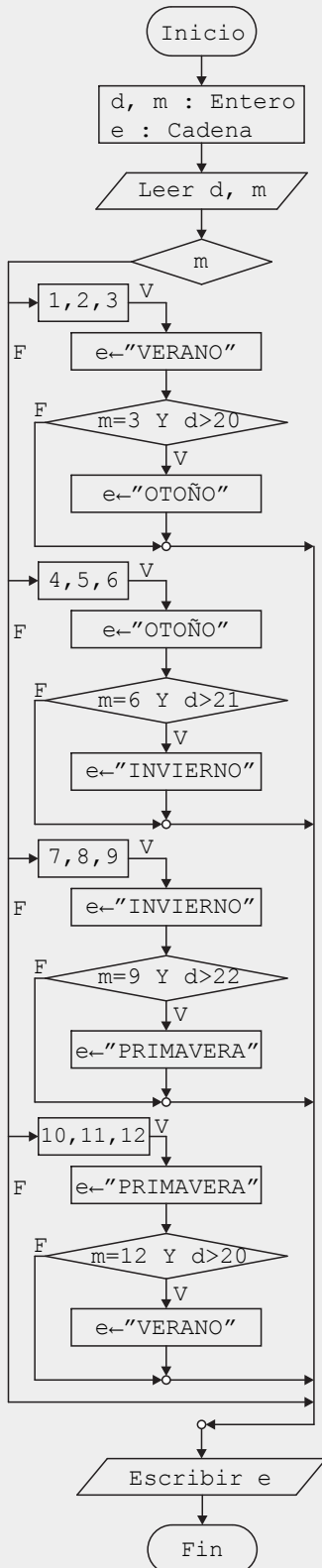
Dia 15

Mes 10

Estacion PRIMAVERA

Calcular

Diagrama de flujo



Algoritmo

Pseudocódigo

Inicio**//Variables**

d, m : Entero
e : Cadena

//Entrada

Leer d, m

//Proceso

En Caso que m Sea

Caso 1, 2, 3

e ← "VERANO"

Si m = 3 Y d > 20 Entonces

e ← "OTOÑO"

Fin Si

Caso 4, 5, 6

e ← "OTOÑO"

Si m = 6 Y d > 21 Entonces

e ← "INVIERNO"

Fin Si

Caso 7, 8, 9

e ← "INVIERNO"

Si m = 9 Y d > 22 Entonces

e ← "PRIMAVERA"

Fin Si

Caso 10, 11, 12

e ← "PRIMAVERA"

Si m = 12 Y d > 20 Entonces

e ← "VERANO"

Fin Si

Fin Caso

//Salida

Escribir e

Fin

Codificación:

```

<?php

//Variables
$d = 0; $m = 0; $e = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $d = (int)$_POST["txttd"];
    $m = (int)$_POST["txtm"];

    //Proceso
    switch($m){
        case 1:
        case 2:
        case 3:
            $e = "VERANO";
            if($m == 3 && $d > 20)
                $e = "OTOÑO";

            break;
        case 4:
        case 5:
        case 6:
            $e = "OTOÑO";
            if($m == 6 && $d > 21)
                $e = "INVIERNO";

            break;
        case 7:
        case 8:
        case 9:
            $e = "INVIERNO";
            if($m == 9 && $d > 22)
                $e = "PRIMAVERA";

            break;
        case 10:
        case 11:
        case 12:
            $e = "PRIMAVERA";
            if($m == 12 && $d > 20)
                $e = "VERANO";

            break;
    }
}
?>
<html>
<head>
<title>Problema 32</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->

```

```

</style>
</head>
<body>
<form method="post" action="Problema32.php">
  <table width="241" border="0">
    <tr>
      <td colspan="2"><strong>Problema 32</strong> </td>
    </tr>
    <tr>
      <td width="81">Dia</td>
      <td width="150">
        <input name="txtD" type="text" id="txtD" value="<?=$d?>" />
      </td>
    </tr>
    <tr>
      <td>Mes</td>
      <td><input name="txtM" type="text" id="txtM" value="<?=$m?>"
    /></td>
    </tr>
    <tr>
      <td>Estacion</td>
      <td><input name="txtE" type="text" class="TextoFondo" id="txtE"
value="<?=$e?>" /></td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td>
        <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
      </td>
    </tr>
  </table>
</form>

```

Problema n.º 33

Enunciado: En una universidad se han establecido los siguientes puntajes de ingreso a sus respectivas facultades.

Facultad	Puntaje mínimo
Sistemas	100
Electrónica	90
Industrial	80
Administración	70

De acuerdo al puntaje obtenido por un postulante, determinar la facultad a la cual ingresó o dar un mensaje correspondiente en caso no ingrese.

Análisis: Para la solución de este problema se requiere que el usuario ingrese el puntaje; luego, que el sistema verifique y devuelva la facultad a la que ingresó.

Entrada

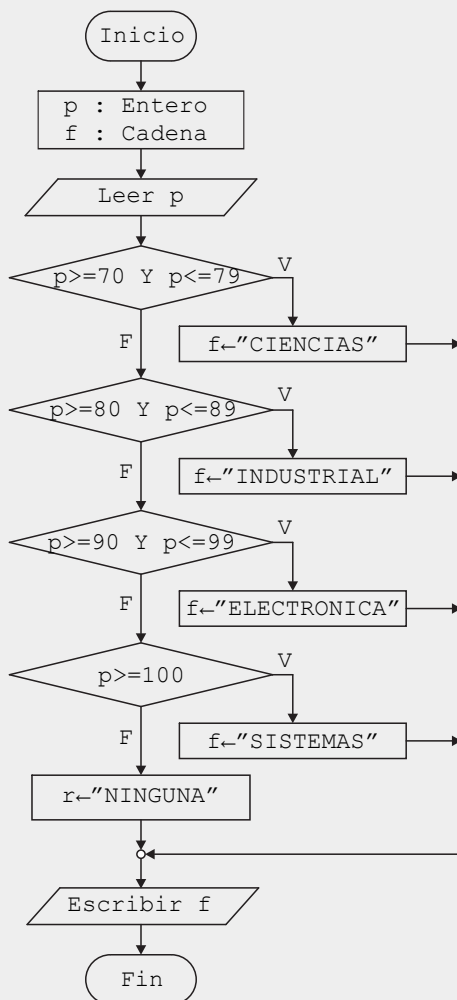
- Puntaje (p)

Salida

- Facultad (f)

Diseño:**Interfaz de usuario**

The screenshot shows a web browser window titled 'Problema 33 - Windows Internet Explorer'. The address bar shows 'http://localhost/Cap04/Problema33.php'. The browser has a menu bar with 'Archivo', 'Edición', 'Ver', 'Favoritos', 'Herramientas', and 'Ayud'. Below the menu is a toolbar with a star icon and a plus icon. The main content area is titled 'Problema 33' and contains two input fields: 'Puntaje' with the value '120' and 'Facultad' with the value 'SISTEMAS'. Below these fields is a button labeled 'Calcular'.

Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio****//Variables**

p : Entero

f : Cadena

//Entrada

Leer p

//Proceso

Si p >= 70 Y p <= 79 Entonces

f ← "CIENCIAS"

SiNo Si p >= 80 Y p <= 89 Entonces

f ← "INDUSTRIAL"

SiNo Si p >= 90 Y p <= 99 Entonces

f ← "ELECTRONICA"

SiNo Si p >= 100 Entonces

f ← "SISTEMAS"

SiNo

f ← "NINGUNO"

Fin Si

//Salida

Escribir f

Fin

Codificación:

```

<?php

//Variables
$p = 0; $f = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $p = (int)$_POST["txtp"];

    //Proceso
    if($p >= 70 && $p <= 79)
        $f = "CIENCIAS";
    elseif($p >= 80 && $p <= 89)
        $f = "INDUSTRIAL";
    elseif($p >= 90 && $p <= 99)
        $f = "ELECTRONICA";
    elseif($p >= 100)
        $f = "SISTEMAS";
    else
        $f = "NINGUNA";
}
?>
<html>
<head>
<title>Problema 33</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema33.php">
    <table width="241" border="0">
        <tr>
            <td colspan="2"><strong>Problema 33</strong> </td>
        </tr>
        <tr>
            <td width="81">Puntaje</td>
            <td width="150">
                <input name="txtp" type="text" id="txtp" value="<?=$p?>" />
            </td>
        </tr>
        <tr>
            <td>Facultad</td>
            <td><input name="txtf" type="text" class="TextoFondo" id="txtf"
                value="<?=$f?>" /></td>
        </tr>
    </table>
</form>

```

```

        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 34

Enunciado: Determine el importe a pagar para el examen de admisión de una universidad, cuyo valor depende del nivel socioeconómico y el colegio de procedencia.

Colegio	Nivel Social		
	A	B	C
Nacional	300	200	100
Particular	400	300	200

Análisis: Para la solución de este problema se requiere que el usuario ingrese el colegio y el nivel socioeconómico al que pertenece; luego, que el sistema verifique y determine el monto a pagar.

Entrada

- Colegio (c)
- Nivel (n)

Salida

- Monto a pagar (mp)

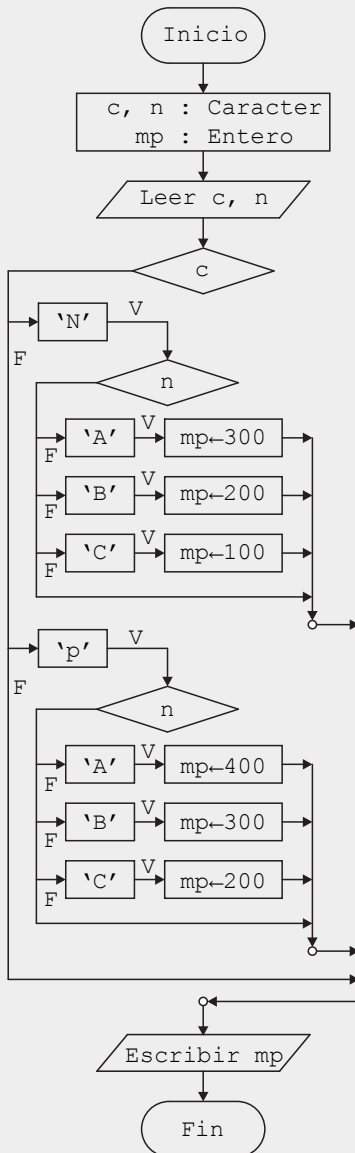
Diseño:

Interfaz de usuario

The screenshot shows a web browser window titled "Problema 34 - Windows Internet Explorer". The address bar shows "http://localhost/Cap04/Problema34.php". The browser menu includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayud". The page content is titled "Problema 34" and contains three input fields: "Colegio" with the value "P", "Nivel" with the value "B", and "Monto a pagar" with the value "300". Below these fields is a button labeled "Calcular".

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

c, n : Carácter
mp : Entero

//Entrada

Leer c, n

//Proceso

En Caso que c Sea

Caso 'N'

En Caso que n Sea

Caso 'A'

mp ← 300

Caso 'B'

mp ← 200

Caso 'C'

mp ← 100

Fin Caso

Caso 'P'

En Caso que n Sea

Caso 'A'

mp ← 400

Caso 'B'

mp ← 300

Caso 'C'

mp ← 200

Fin Caso

Fin Caso

//Salida

Escribir mp

Fin

Codificación:

```

<?php

//Variables
$c = ''; $n = ''; $mp = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $c = $_POST["txtc"];
    $n = $_POST["txtn"];

    //Proceso
    if($c == 'N'){
        if($n == 'A')
            $mp = 300;
        elseif($n == 'B')
            $mp = 200;
        elseif($n == 'C')
            $mp = 100;
    }elseif($c == 'P'){
        if($n == 'A')
            $mp = 400;
        elseif($n == 'B')
            $mp = 300;
        elseif($n == 'C')
            $mp = 200;
    }
}

?>
<html>
<head>
<title>Problema 34</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema34.php">
    <table width="288" border="0">
        <tr>
            <td colspan="2"><strong>Problema 34</strong> </td>
        </tr>
        <tr>
            <td width="87">Colegio</td>
            <td width="144">
                <input name="txtc" type="text" id="txtc" value="<?=$c?>" />
            </td>
        </tr>
        <tr>
            <td>Nivel</td>

```

```

        <td><input name="txtn" type="text" id="txtn" value="<?=$n?>"
/></td>
    </tr>

    <tr>
        <td>Monto a pagar </td>
        <td><input name="txtmp" type="text" class="TextoFondo" id="txtmp"
value="<?=$mp?>" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 35

Enunciado: Dado el número del mes y el año (cuatro dígitos) de una fecha, determinar, en letras, el nombre del mes y cuantos días tiene. Considerar que febrero tiene 28 o 29 días si el año es bisiesto, un año es bisiesto si es múltiplo de 4, pero no de 100 y sí de 400.

Análisis: Para la solución de este problema se requiere que el usuario ingrese el número del mes y el año; luego, que el sistema verifique y determine cuántos días tiene y el nombre del mes.

Entrada

- Mes (m)
- Año (a)

Salida

- Días (d)
- Mes letras (ml)

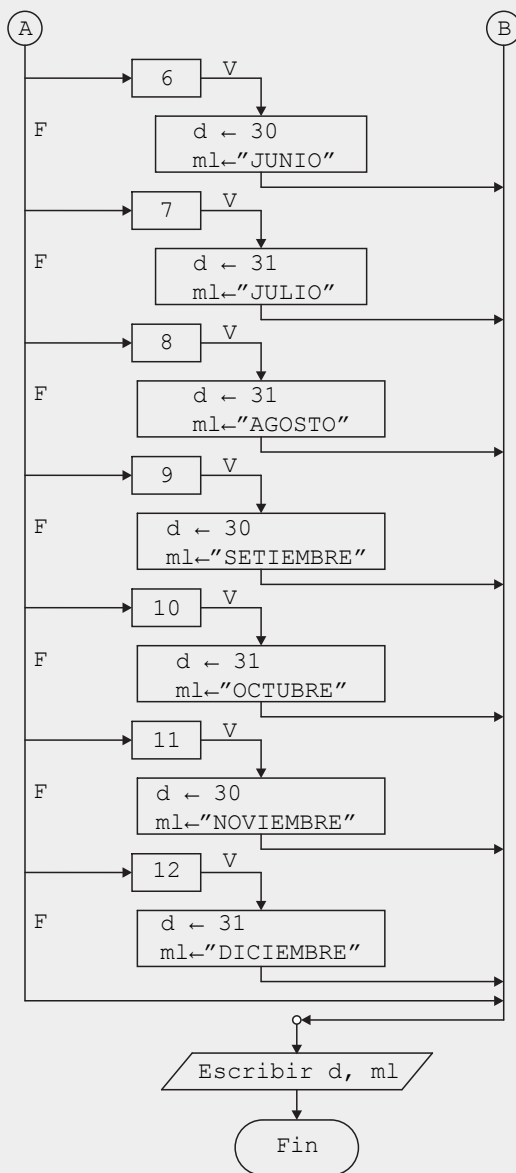
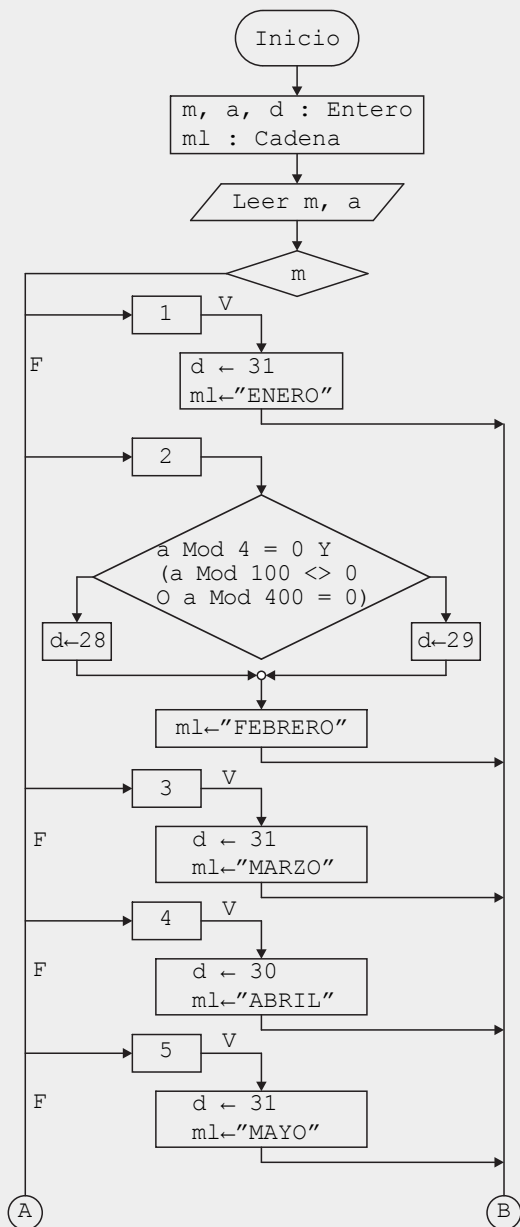
Diseño:

Interfaz de usuario

The screenshot shows a web browser window titled "Problema 35 - Windows Internet Explorer". The address bar shows "http://localhost/Cap04/Problema35.php". The browser has a menu bar with "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". Below the menu bar is a toolbar with "Problema 35". The main content area displays the form for "Problema 35". The form has four input fields: "Mes" with the value "2", "Año" with the value "2008", "Mes" with the value "FEBRERO", and "Días" with the value "29". Below these fields is a "Calcular" button.

Algoritmo

Diagrama de flujo



Pseudocódigo**Inicio****//Variables**

```
m, a, d : Entero
ml : Cadena
```

//Entrada

```
Leer m, a
```

//Proceso

```
En Caso que m Sea
```

```
  Caso 1
```

```
    d ← 31
```

```
    ml ← "ENERO"
```

```
  Caso 2
```

```
    Si a Mod 4 = 0 Y (a Mod 100 <> 0 O
                                a Mod 400 = 0) Entonces
```

```
      d ← 29
```

```
    SiNo
```

```
      d ← 28
```

```
    Fin Si
```

```
    ml ← "FEBRERO"
```

```
  Caso 3
```

```
    d ← 31
```

```
    ml ← "MARZO"
```

```
  Caso 4
```

```
    d ← 30
```

```
    ml ← "ABRIL"
```

```
  Caso 5
```

```
    d ← 31
```

```
    ml ← "MAYO"
```

```
  Caso 6
```

```
    d ← 30
```

```
    ml ← "JUNIO"
```

```
  Caso 7
```

```
    d ← 31
```

```
    ml ← "JULIO"
```

```
  Caso 8
```

```
    d ← 31
```

```
    ml ← "AGOSTO"
```

```
  Caso 9
```

```
    d ← 30
```

```
    ml ← "SEPTIEMBRE"
```

```
  Caso 10
```

```
    d ← 31
```

```
    ml ← "OCTUBRE"
```

```
  Caso 11
```

```
    d ← 30
```

```
    ml ← "NOVIEMBRE"
```

```
  Caso 12
```

```
    d ← 31
```

```
    ml ← "DICIEMBRE"
```

```
Fin Caso
```

//Salida

```
Escribir d, ml
```

Fin

Codificación:

```
<?php

//Variables
$m = 0; $a = 0; $d = 0; $ml = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $m = (int)$ _POST["txtm"];
    $a = (int)$ _POST["txta"];

    //Proceso
    switch($m){
        case 1:
            $d = 31;
            $ml = "ENERO";
            break;
        case 2:
            if($a % 4 == 0 && ($a % 100 != 0 || $a % 400 == 0))
                $d = 29;
            else
                $d = 28;

            $ml = "FEBRERO";
            break;
        case 3:
            $d = 31;
            $ml = "MARZO";
            break;
        case 4:
            $d = 30;
            $ml = "ABRIL";
            break;
        case 5:
            $d = 31;
            $ml = "MAYO";
            break;
        case 6:
            $d = 30;
            $ml = "JUNIO";
            break;
        case 7:
            $d = 31;
            $ml = "JULIO";
            break;
        case 8:
            $d = 31;
            $ml = "AGOSTO";
            break;
        case 9:
            $d = 30;
            $ml = "SEPTIEMBRE";
            break;
        case 10:
            $d = 31;
            $ml = "OCTUBRE";
            break;
        case 11:
            $d = 30;
            $ml = "NOVIEMBRE";
```

```

        break;
    case 12:
        $d = 31;
        $m1 = "DICIEMBRE";
        break;
    }
}
?>
<html>
<head>
<title>Problema 35</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema35.php">
    <table width="288" border="0">
        <tr>
            <td colspan="2"><strong>Problema 35 </strong></td>
        </tr>
        <tr>
            <td width="87">Mes</td>
            <td width="144">
                <input name="txtm" type="text" id="txtm" value="<?=$m?>" />
            </td>
        </tr>
        <tr>
            <td>A&ntilde;o</td>
            <td><input name="txta" type="text" id="txta" value="<?=$a?>" /></td>
        </tr>

        <tr>
            <td>Mes</td>
            <td><input name="txtm1" type="text" class="TextoFondo" id="txtm1" value="<?=$m1?>" /></td>
        </tr>
        <tr>
            <td>Dias</td>
            <td><input name="txtd" type="text" class="TextoFondo" id="txtd" value="<?=$d?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular" value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 36

Enunciado: Una empresa ha establecido diferentes precios a sus productos, según la calidad.

Producto \ Calidad	1	2	3
	1	2	3
1	5000	4500	4000
2	4500	4000	3500
3	4000	3500	3000

Cree un programa que devuelva el precio a pagar por un producto y una calidad dada.

Análisis: Para la solución de este problema se requiere que el usuario ingrese la calidad y el producto; luego, que el sistema verifique y determine el precio.

Entrada

- Calidad (c)
- Producto (p)

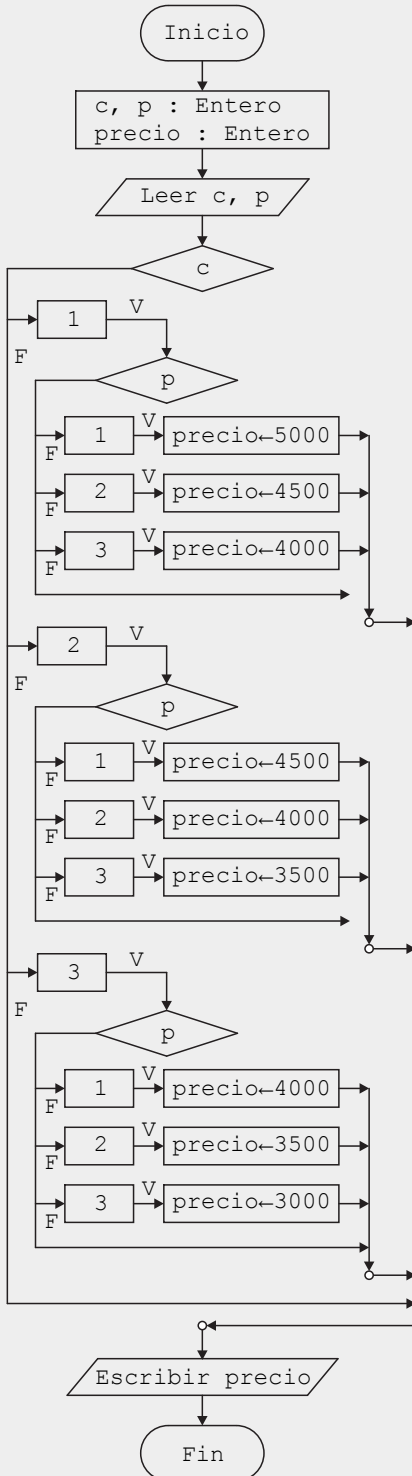
Salida

- Precio (precio)

Diseño:**Interfaz de usuario**

The screenshot shows a web browser window titled "Problema 36 - Windows Internet Explorer". The address bar shows "http://localhost/Cap04/Problema36.php". The browser menu includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The page content is titled "Problema 36" and contains three input fields: "Calidad" with the value "2", "Producto" with the value "3", and "Precio" with the value "3500". Below these fields is a "Calcular" button.

Algoritmo



Pseudocódigo

Inicio

```
//Variables
```

```
c, p : Entero
precio : Entero
```

```
//Entrada
```

Leer c, p

```
//Proceso
```

En Caso que c Sea

Caso 1

En Caso que p Sea

Caso 1

```
precio ← 5000
```

Caso 2

```
precio ← 4500
```

Caso 3

```
precio ← 4000
```

Fin Caso

Caso 2

En Caso que p Sea

Caso 1

```
precio ← 4500
```

Caso 2

```
precio ← 4000
```

Caso 3

```
precio ← 3500
```

Fin Caso

Caso 3

En Caso que p Sea

Caso 1

```
precio ← 4000
```

Caso 2

```
precio ← 3500
```

Caso 3

```
precio ← 3000
```

Fin Caso

Fin Caso

```
//Salida
```

Escribir precio

Fin

Codificación:

```

<?php

//Variables
$c = 0; $p = 0; $precio = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $c = (int)$_POST["txtc"];
    $p = (int)$_POST["txtp"];

    //Proceso
    switch($c){
        case 1:
            switch($p){
                case 1:
                    $precio = 5000;
                    break;
                case 2:
                    $precio = 4500;
                    break;
                case 3:
                    $precio = 4000;
                    break;
            }
            break;
        case 2:
            switch($p){
                case 1:
                    $precio = 4500;
                    break;
                case 2:
                    $precio = 4000;
                    break;
                case 3:
                    $precio = 3500;
                    break;
            }
            break;
        case 3:
            switch($p){
                case 1:
                    $precio = 4000;
                    break;
                case 2:
                    $precio = 3500;
                    break;
                case 3:
                    $precio = 3000;
                    break;
            }
    }
}

?>
<html>
<head>
<title>Problema 36</title>
<style type="text/css">
<!--
.TextoFondo {

```

```

        background-color: #CCFFFF;
    }
-->
</style>
</head>
<body>
<form method="post" action="Problema36.php">
    <table width="288" border="0">
        <tr>
            <td colspan="2"><strong>Problema 36</strong> </td>
        </tr>
        <tr>
            <td width="87">Calidad</td>
            <td width="144">
                <input name="txtc" type="text" id="txtc" value="<?=$c?>" />
            </td>
        </tr>
        <tr>
            <td>Producto</td>
            <td><input name="txtp" type="text" id="txtp" value="<?=$p?>" /></td>
        </tr>
        <tr>
            <td>Precio</td>
            <td><input name="txtprecio" type="text" class="TextoFondo" id="txtprecio" value="<?=$precio?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular" value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 37

Enunciado: Diseñe un algoritmo que califique el puntaje obtenido en el lanzamiento de tres dados en base a la cantidad de seis obtenidos, de acuerdo a lo siguiente:

Tres seis: Oro

Dos seis: Plata

Un seis: Bronce

Ningún seis: Perdió

Análisis: Para la solución de este problema se requiere que el usuario ingrese el puntaje de los dados; luego, que el sistema verifique y determine el premio.

Entrada

- Primer dado (d1)
- Segundo dado (d2)
- Tercer dado (d3)

Salida

- Premio (p)

Diseño:**Interfaz de usuario**

Problema 37 - Windows Internet Explorer

http://localhost/Cap04/Problema37.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 37

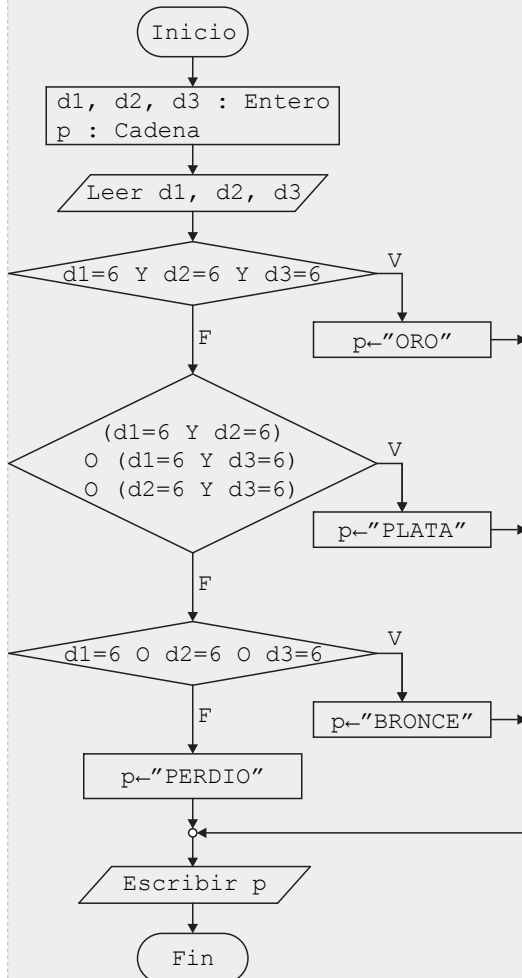
Problema 37

Dado1

Dado2

Dado3

Premio

Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio****//Variables**

d1, d2, d3 : Entero
p : Cadena

//Entrada

Leer d1, d2, d3

//Proceso

Si d1=6 Y d2=6 Y d3=6 Entonces

p ← "ORO"

SiNoSi (d1=6 Y d2=6) O (d1=6 Y d3=6)
O (d2=6 Y d3=6) Entonces

p ← "PLATA"

SiNoSi d1=6 O d2=6 O d3=6 Entonces
p ← "BRONCE"

SiNo

p ← "PERDIO"

Fin Si

//Salida

Escribir p

Fin

Codificación:

```

<?php

//Variables
$d1 = 0; $d2 = 0; $d3 = 0; $p = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $d1 = (int)$_POST["txtd1"];
    $d2 = (int)$_POST["txtd2"];
    $d3 = (int)$_POST["txtd3"];

    //Proceso
    if($d1 == 6 && $d2 == 6 && $d3 == 6)
        $p = "ORO";
    elseif(($d1 == 6 && $d2 == 6) || ($d1 == 6 && $d3 == 6) &&
        ($d2 == 6 && $d3 == 6))
        $p = "PLATA";
    elseif($d1 == 6 || $d2 == 6 || $d3 == 6)
        $p = "BRONCE";
    else
        $p = "PERDIO";
}
?>
<html>
<head>
<title>Problema 37</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema37.php">
    <table width="288" border="0">
        <tr>
            <td colspan="2"><strong>Problema 37</strong> </td>
        </tr>
        <tr>
            <td width="87">Dado1</td>
            <td width="144">
                <input name="txtd1" type="text" id="txtd1" value="<?=$d1?>"
            />
            </td>
        </tr>
        <tr>
            <td>Dado2</td>
            <td><input name="txtd2" type="text" id="txtd2" value="<?=$d2?>"
        /></td>
        </tr>

        <tr>
            <td>Dado3</td>

```

```

        <td><input name="txtD3" type="text" id="txtD3" value="<?=$d3?>"
/></td>
    </tr>

    <tr>
        <td>Premio</td>
        <td><input name="txtp" type="text" class="TextoFondo" id="txtp"
value="<?=$p?>" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 38

Enunciado: Dado el día, mes y año, determine si es una fecha correcta. Considere los años bisiestos.

Análisis: Para la solución de este problema se requiere que el usuario ingrese el día, mes y año; luego, que el sistema verifique y determine si el resultado es o no una fecha correcta.

Entrada

- Día (d)
- Mes (m)
- Año (a)

Salida

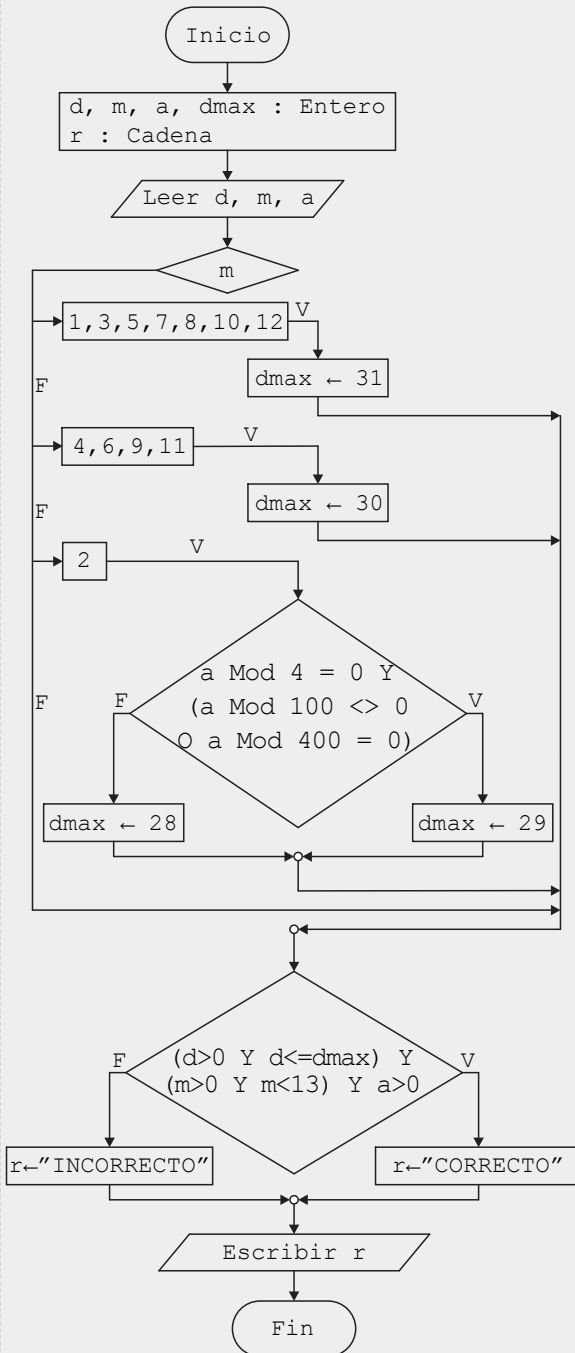
- Respuesta (r)

Diseño:

Interfaz de usuario

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

d, m, a, dmax : Entero
r : Cadena

//Entrada

Leer d, m, a

//Proceso

En Caso que m Sea

Caso 1, 3, 5, 7, 8, 10, 12

dmax ← 31

Caso 4, 6, 9, 11

dmax ← 30

Caso 2

Si a Mod 4 = 0 Y (a Mod 100 <> 0
O a Mod 400 = 0) Entonces

dmax ← 29

SiNo

dmax ← 28

Fin Si

Fin Caso

Si d>0 Y d<=dmax) Y (m>0 Y m<13)
Y a>0 Entonces

r ← "CORRECTO"

SiNo

r ← "INCORRECTO"

Fin Si

//Salida

Escribir r

Fin

Codificación:

```

<?php

//Variables
$d = 0; $m = 0; $a = 0; $dmax = 0;
$r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $d = (int)$_POST["txtd"];
    $m = (int)$_POST["txtm"];
    $a = (int)$_POST["txta"];

    //Proceso
    switch($m) {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            $dmax = 31;
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            $dmax = 30;
            break;
        case 2:
            if($a % 4 == 0 && (!$a % 100 == 0) || $a % 400 == 0)
                $dmax = 29;
            else
                $dmax = 28;
    }

    if (($d > 0 && $d <= $dmax) && ($m > 0 && $m < 13) && $a > 0)
        $r = "CORRECTO";
    else
        $r = "INCORRECTO";
}
?>
<html>
<head>
<title>Problema 38</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema38.php">
    <table width="288" border="0">
        <tr>
            <td colspan="2"><strong>Problema 38</strong> </td>
        </tr>
        <tr>
            <td width="87">Dia</td>
            <td width="144">
                <input name="txtd" type="text" id="txtd" value="<?=$d?>" />
            </td>
        </tr>
        <tr>

```

```

        <td>Mes</td>
        <td><input name="txtm" type="text" id="txtm" value="<?=$m?>" /></td>
    </tr>

    <tr>
        <td>A&ntilde;o</td>
        <td><input name="txta" type="text" id="txta" value="<?=$a?>" /></td>
    </tr>

    <tr>
        <td>Respuesta</td>
        <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 39

Enunciado: Dada una fecha válida, halle la fecha del siguiente día.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese el día, mes y año; luego, que el sistema devuelve la fecha del siguiente día.

Entrada

- Día (d)
- Mes (m)
- Año (a)

Salida

- Día (d)
- Mes (m)
- Año (a)

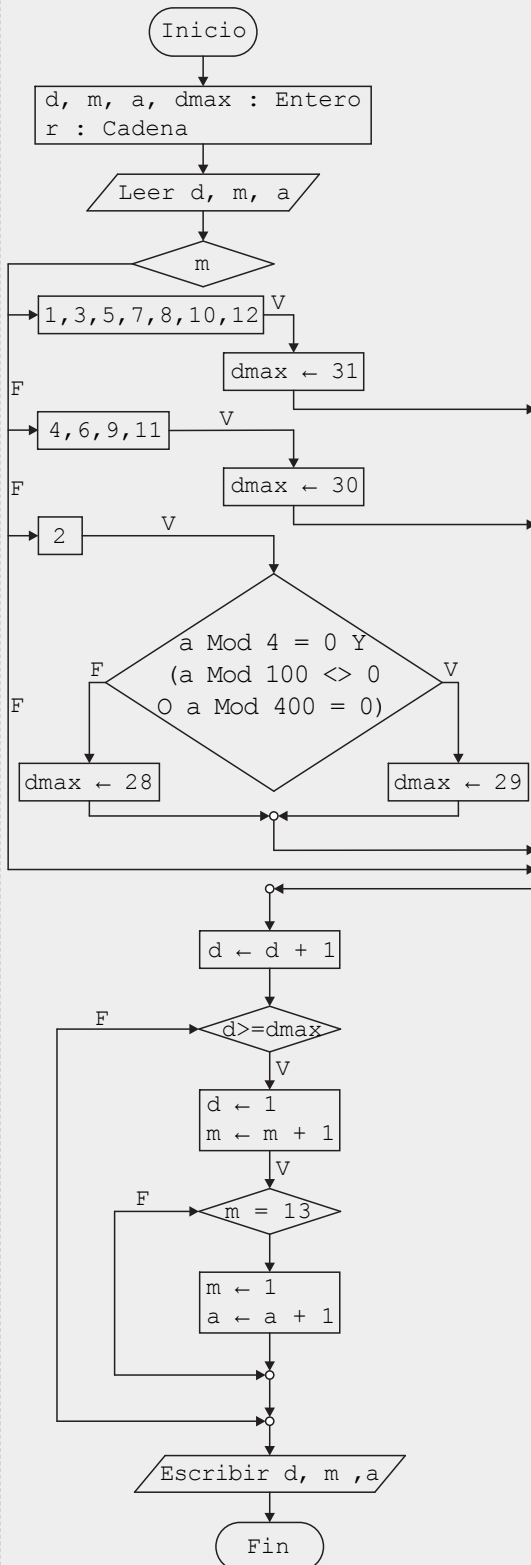
Diseño:

Interfaz de usuario

The screenshot shows a web browser window titled "Problema 39 - Windows Internet Explorer". The address bar displays "http://localhost/Cap04/Problema39.php". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The page content is titled "Problema 39" and features a form with three input fields: "Día" (containing "31"), "Mes" (containing "12"), and "Año" (containing "2008"). A "Calcular" button is positioned below the "Año" field.

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

d, m, a, dmax : Entero
r : Cadena

//Entrada

Leer d, m, a

//Proceso

En Caso que m Sea

Caso 1,3,5,7,8,10,12

dmax ← 31

Caso 4, 6, 9, 11

dmax ← 30

Caso 2

Si a Mod 4 = 0 And (a Mod 100 <> 0
Or a Mod 400 = 0) Entonces

dmax ← 29

SiNo

dmax ← 28

Fin Si

Fin Caso

d = d + 1

Si d > dmax Entonces

d ← 1

m ← m + 1

Si m = 13 Entonces

m ← 1

a ← a + 1

Fin Si

Fin Si

//Salida

Escribir d, m, a

Fin

Codificación:

```

<?php

//Variables
$d = 0; $m = 0; $a = 0; $dmax = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $d = (int)$_POST["txtd"];
    $m = (int)$_POST["txtm"];
    $a = (int)$_POST["txta"];

    //Proceso
    switch($m) {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            $dmax = 31;
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            $dmax = 30;
            break;
        case 2:
            if($a % 4 == 0 && (!($a % 100 == 0) || $a % 400 == 0))
                $dmax = 29;
            else
                $dmax = 28;
    }

    $d = $d + 1;

    if($d > $dmax) {
        $d = 1;
        $m +=1;
        if($m == 13){
            $m = 1;
            $a++;
        }
    }
}
?>
<html>
<head>
<title>Problema 39</title>
</head>
<body>
<form method="post" action="Problema39.php">
    <table width="288" border="0">
        <tr>
            <td colspan="2"><strong>Problema 39</strong> </td>
        </tr>
        <tr>
            <td width="87">Dia</td>

```

```

        <td width="144">
            <input name="txtd" type="text" id="txtd" value="<?=$d?>" />
        </td>
    </tr>
    <tr>
        <td>Mes</td>
        <td><input name="txtm" type="text" id="txtm" value="<?=$m?>"
    /></td>
    </tr>

    <tr>
        <td>A&ntilde;o</td>
        <td><input name="txta" type="text" id="txta" value="<?=$a?>"
    /></td>
    </tr>

    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
            value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 40

Enunciado: Convierta a números romanos, números menores a 4000.

Análisis: Para la solución de este problema se requiere que el usuario ingrese el número; luego, que el sistema convierta y devuelva el número en romano.

Entrada

- Número decimal (n)

Salida

- Número romano (r)

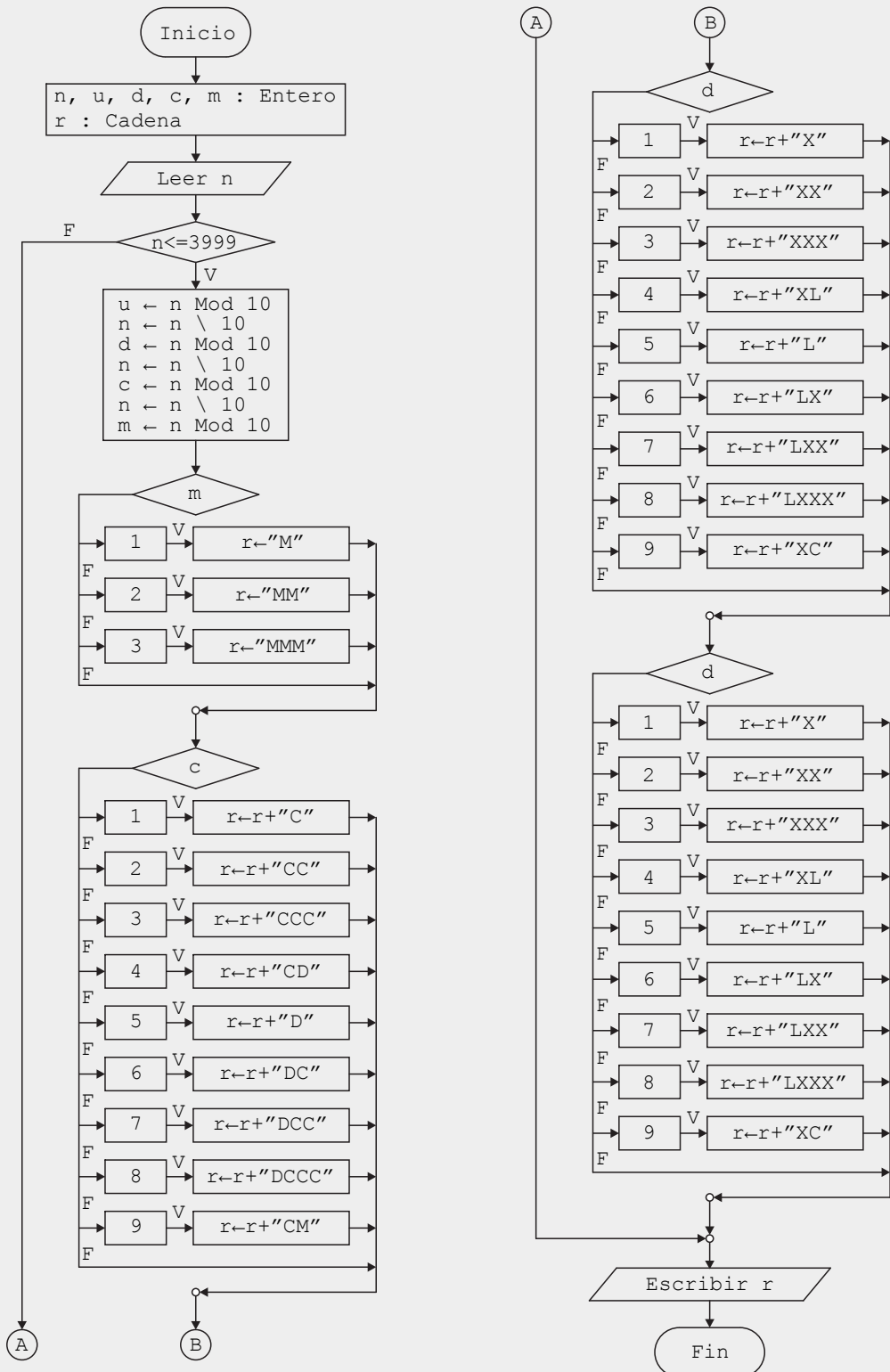
Diseño:

Interfaz de usuario

The screenshot shows a web browser window titled "Problema 40 - Windows Internet Explorer". The address bar displays "http://localhost/Cap04/Problema40.php". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayud". The page content is titled "Problema 40" and contains a form with two input fields: "Numero" (containing "1150") and "Romano" (containing "MCL"). Below these fields is a "Calcular" button.

Algoritmo

Diagrama de flujo



Pseudocódigo**Inicio****//Variables**

```
n, u, d, c, m : Entero  
r : Cadena
```

//Entrada

```
Leer n
```

//Proceso

```
Si n <= 3999 Entonces
```

```
    u ← n Mod 10  
    n ← n \ 10  
    d ← n Mod 10  
    n ← n \ 10  
    c ← n Mod 10  
    n ← n \ 10  
    m ← n Mod 10
```

```
    En Caso que m Sea
```

```
        Caso 1
```

```
            r ← "M"
```

```
        Caso 2
```

```
            r ← "MM"
```

```
        Caso 3
```

```
            r ← "MMM"
```

```
    Fin Caso
```

```
    En Caso que c Sea
```

```
        Caso 1
```

```
            r ← r + "C"
```

```
        Caso 2
```

```
            r ← r + "CC"
```

```
        Caso 3
```

```
            r ← r + "CCC"
```

```
        Caso 4
```

```
            r ← r + "CD"
```

```
        Caso 5
```

```
            r ← r + "D"
```

```
        Caso 6
```

```
            r ← r + "DC"
```

```
        Caso 7
```

```
            r ← r + "DCC"
```

```
        Caso 8
```

```
            r ← r + "DCCC"
```

```
        Caso 9
```

```
            r ← r + "CM"
```

```
    Fin Caso
```

```
    En Caso que d Sea
```

```
Caso 1
    r ← r + "X"
Caso 2
    r ← r + "XX"
Caso 3
    r ← r + "XXX"
Caso 4
    r ← r + "XL"
Caso 5
    r ← r + "L"
Caso 6
    r ← r + "LX"
Caso 7
    r ← r + "LXX"
Caso 8
    r ← r + "LXXX"
Caso 9
    r ← r + "XC"
Fin Caso
En Caso que u Sea
    Caso 1
        r ← r + "I"
    Caso 2
        r ← r + "II"
    Caso 3
        r ← r + "III"
    Caso 4
        r ← r + "IV"
    Caso 5
        r ← r + "V"
    Caso 6
        r ← r + "VI"
    Caso 7
        r ← r + "VII"
    Caso 8
        r ← r + "VIII"
    Caso 9
        r ← r + "IX"
    Fin Caso
Fin Si
```

```
//Salida
Escribir r
```

```
Fin
```

Codificación:

```
<?php

//Variables
$n = 0; $u = 0; $d = 0; $c = 0; $m = 0; $nn = 0;
$r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $nn = $n;
    if ($n <= 3999) {
        $u = $n % 10;
        $n = (int)($n / 10);
        $d = $n % 10;
        $n = (int)($n / 10);
        $c = $n % 10;
        $n = (int)($n / 10);
        $m = $n % 10;
        $n = $nn;
        switch ($m){
            case 1:
                $r = "M";
                break;
            case 2:
                $r = "MM";
                break;
            case 3:
                $r = "MMM";
                break;
        }

        switch ($c){
            case 1:
                $r = $r . "C";
                break;
            case 2:
                $r = $r . "CC";
                break;
            case 3:
                $r = $r . "CCC";
                break;
            case 4:
                $r = $r . "CD";
                break;
            case 5:
                $r = $r . "D";
                break;
            case 6:
                $r = $r . "DC";
                break;
            case 7:
                $r = $r . "DCC";
                break;
        }
    }
}
```

```
        case 8:
            $r = $r . "DCCC";
            break;
        case 9:
            $r = $r . "CM";
            break;
    }

    switch ($d){
        case 1:
            $r = $r . "X";
            break;
        case 2:
            $r = $r . "XX";
            break;
        case 3:
            $r = $r . "XXX";
            break;
        case 4:
            $r = $r . "XL";
            break;
        case 5:
            $r = $r . "L";
            break;
        case 6:
            $r = $r . "LX";
            break;
        case 7:
            $r = $r . "LXX";
            break;
        case 8:
            $r = $r . "LXXX";
            break;
        case 9:
            $r = $r . "XC";
            break;
    }

    switch($u){
        case 1:
            $r = $r . "I";
            break;
        case 2:
            $r = $r . "II";
            break;
        case 3:
            $r = $r . "III";
            break;
        case 4:
            $r = $r . "IV";
            break;
        case 5:
            $r = $r . "V";
            break;
        case 6:
            $r = $r . "VI";
            break;
    }
```

```

        case 7:
            $r = $r . "VII";
            break;
        case 8:
            $r = $r . "VIII";
            break;
        case 9:
            $r = $r . "IX";
            break;
    }
}
}
?>
<html>
<head>
<title>Problema 40</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema40.php">
    <table width="238" border="0">
        <tr>
            <td colspan="2"><strong>Problema 40 </strong></td>
        </tr>
        <tr>
            <td width="81" height="29">Numero</td>
            <td width="197">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Romano</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

4.3 Problemas propuestos

Los siguientes problemas le servirán para medir su aprendizaje, es importante que los resuelva.

Propuesto n.º 21

Enunciado: Dado el número de un mes, devolver el mes en letras.

Propuesto n.º 22

Enunciado: Lea un número del 1 al 7 y devuelva el día de la semana, considere que 1 es domingo.

Propuesto n.º 23

Enunciado: Dado los siguientes operadores aritméticos +, -, * y / , devuelva el nombre del operador.

Propuesto n.º 24

Enunciado: Dado el número de un canal de televisión, determine cuál es el nombre del canal.

Propuesto n.º 25

Enunciado: En una empresa se ha determinado la siguiente política de descuento:

Tarjeta \ Sexo	Hombres	Mujeres
Obrero	15 %	10 %
Empleado	20 %	15 %

Determine mediante un programa, cual será el monto del descuento al sueldo ingresado de un trabajador.

Propuesto n.º 26

Enunciado: Una frutería ofrece las manzanas con descuento según la siguiente tabla:

Kilos	% Descuento
0- 2	0 %
2.01- 5	10 %
5.01- 10	20 %
Mayor a 10	30 %

Determine cuánto pagará una persona que compra manzanas en esa frutería.

Propuesto n.º 27

Enunciado: Obtenga el nombre del estado civil según la siguiente tabla:

Código	Estado civil
0	Soltero
1	Casado
2	Divorciado
3	Viudo

Propuesto n.º 28

Enunciado: Determinar el monto que recibirá un trabajador por utilidades, después de ingresar el tiempo de servicio y el cargo, según la siguiente tabla:

Tiempo de servicio \ Cargo	Administrador	Contador	Empleado
Entre 0 y 2 años	2000	1500	1000
Entre 3 y 5 años	2500	2000	1500
Entre 6 y 8 años	3000	2500	2000
Mayor a 8 años	4000	3500	1500

Propuesto n.º 29

Enunciado: Según la siguiente tabla, obtener la ciudad que visitará, después de ingresar su sexo y el puntaje obtenido en un examen.

Puntaje \ Sexo	Masculino	Femenino
Entre 18 y 35	Arequipa	Cuzco
Entre 36 y 75	Cuzco	Iquitos
Mayor a 75	Iquitos	Arequipa

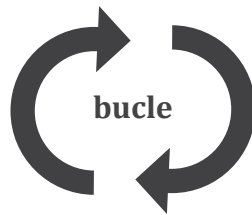
Propuesto n.º 30

Enunciado: Dada una fecha, determine cuántos días faltan para que acabe el año.

Estructura repetitiva «Mientras»

5.1 Introducción

Muchas veces se requiere repetir una o varias instrucciones para llevar a cabo una tarea; en la programación se cuenta con estructuras que permiten realizar este proceso, llamadas también: bucles, iterativas, lazos, entre otros.



Dependiendo del lenguaje de programación, estas incorporan dos o más estructuras repetitivas, dentro de las cuales las infaltables son mientras (**while**) y para (**for**), con las cuales se puede resolver todo problema que involucre procesos repetitivos.

Cuando se trabaja con estas estructuras se utiliza términos como: contadores, acumuladores, forzar la salida del bucle y continuar al inicio del bucle.

5.2 Contador

Son variables enteras que se incrementan (+) o decrementan (-) con un valor constante, por ejemplo, una variable «c», cuyo valor se incrementa de 1 en 1; se conoce como variable «contador».

Ejemplos pseudocódigo

```
c ← c + 1  
i ← i + 2  
j ← j - 1
```

PHP

```
$c = $c + 1;  
$i += 2;  
$j--;
```

5.3 Acumulador

Son variables de cualquier tipo que almacenan valores variables; por ejemplo, la variable «c», cuyo valor se incrementa por el valor que va tomando otra variable llamada «x».

Ejemplo pseudocódigo

```
c ← c + x  
i ← i + c  
j ← j - i
```

PHP

```
$c = $c + $x;  
$i += $c;  
$j -= $i;
```

5.4 Salir del bucle

Es una instrucción que permite forzar la salida de un bucle, para esto los lenguajes de programación incorporan una instrucción que permita realizar dicha operación.

Pseudocódigo

```
Salir
```

PHP

```
break;
```

5.5 Continuar al inicio del bucle

Es una instrucción que permite saltar al inicio del bucle para volver a ejecutarse, para esto los lenguajes de programación incorporan una instrucción que permita realizar dicha operación.

Pseudocódigo

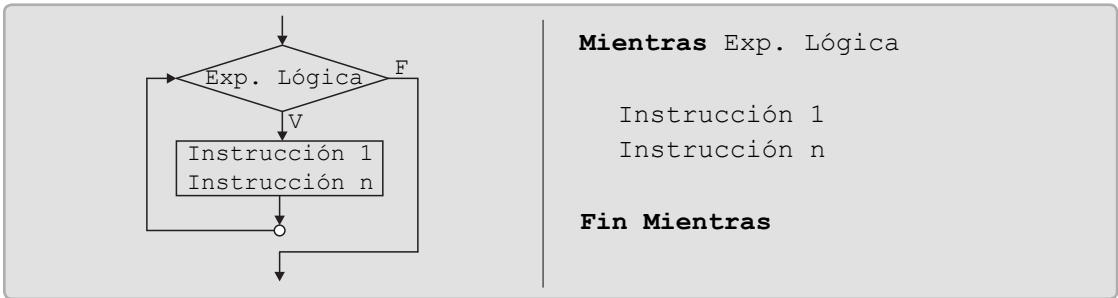
```
Continuar
```

PHP

```
continue;
```

5.6 Estructura repetitiva «Mientras»

Permite repetir una o más instrucciones hasta que la condición (expresión lógica) sea verdadera; cuando la condición es falsa sale del bucle.

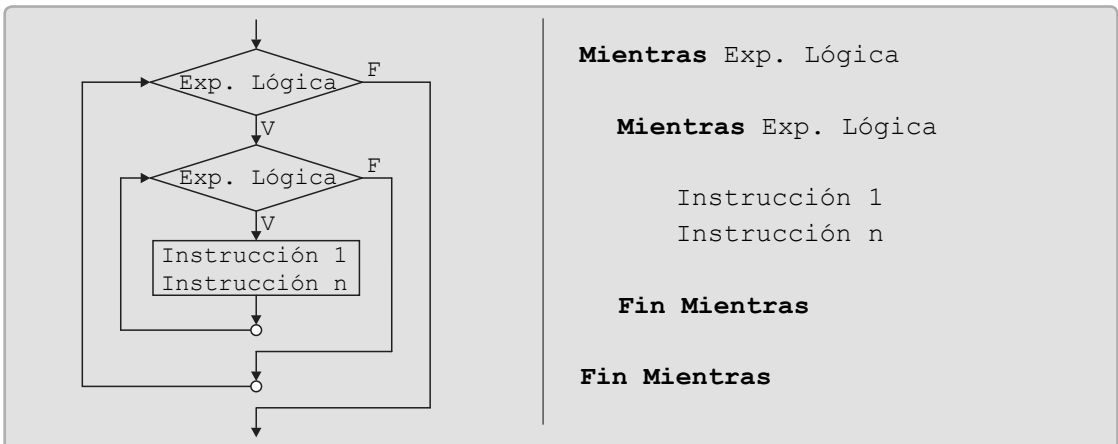


Sintaxis PHP

```
while (<Exp. Log.>) {
    <instrucción 1>;
    <instrucción n>;
}
```

5.7 Estructura repetitiva «Mientras» anidada

Dentro de la estructura repetitiva es posible colocar una o más estructuras repetitivas, así como otras estructuras.



Sintaxis PHP

```
while (<Exp. Log.>) {
    while (<Exp. Log.>) {
        <instruccion1>;
        <instruccionn>;
    }
}
```

Problema n.º 41

Enunciado: Obtener la suma de los primeros N números naturales positivos.

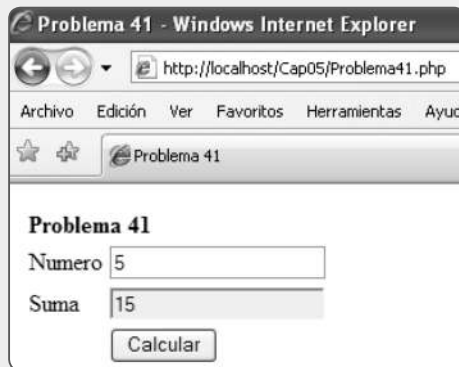
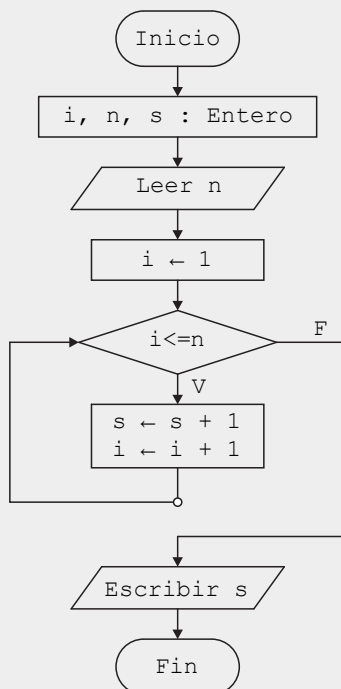
Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema realice el proceso para devolver la suma de los N primeros números.

Entrada

- Número (n)

Salida

- Suma (s)

Diseño:**Interfaz de usuario****Algoritmo****Diagrama de flujo****Pseudocódigo****Inicio****//Variables**

i, n, s : Entero

//Entrada

Leer n

//Proceso

i ← 1

Mientras i ≤ n

 s ← s + 1

 i ← i + 1

Fin Mientras

//Salida

Escribir s

Fin

Codificación:

```

<?php

//Variables
$i = 0;$n = 0;$s = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $i = 1;
    while($i <= $n){
        $s = $s + $i;
        $i = $i + 1;
    }
}
?>
<html>
<head>
<title>Problema 41</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema41.php">
    <table width="222" border="0">
        <tr>
            <td colspan="2"><strong>Problema 41</strong> </td>
        </tr>
        <tr>
            <td width="55">Numero</td>
            <td width="223">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Suma</td>
            <td><input name="txts" type="text" class="TextoFondo" id="txts"
value="<?=$s?>" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 42

Enunciado: Dado un rango de números enteros, obtener la cantidad de números enteros que contiene.

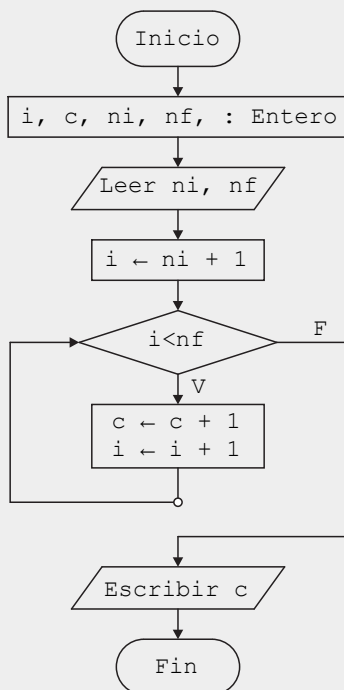
Análisis: Para la solución de este problema, se requiere que el usuario ingrese el número inicial y final; luego, que el sistema procese y devuelva la cantidad de números enteros que contiene el rango.

Entrada

- Número inicial (ni)
- Número final (nf)

Salida

- Cantidad (c)

Diseño:**Interfaz de usuario**
Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio**

//Variables

i, c, ni, nf : Entero

//Entrada

Leer ni, nf

//Proceso

i ← ni + 1

Mientras i < nf

 c ← c + 1

 i ← i + 1

Fin Mientras

//Salida

Escribir c

Fin

Codificación:

```

<?php

//Variables
$i = 0;$ni = 0;$nf = 0; $c = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $ni = (int)$_POST["txttni"];
    $nf = (int)$_POST["txttnf"];

    //Proceso
    $i = $ni + 1;
    while($i < $nf){
        $c +=1;
        $i++;
    }
}
?>
<html>
<head>
<title>Problema 42</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema42.php">
    <table width="288" border="0">
        <tr>
            <td colspan="2"><strong>Problema 42</strong> </td>
        </tr>
        <tr>
            <td width="87">Num. Inicial </td>
            <td width="144">
                <input name="txttni" type="text" id="txttni" value="<?=$ni?" />
            </td>
        </tr>
        <tr>
            <td>Num. Final </td>
            <td><input name="txttnf" type="text" id="txttnf" value="<?=$nf?" /></td>
        </tr>

        <tr>
            <td>Cantidad</td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 43

Enunciado: Dado un rango de números enteros, obtener la cantidad de números pares que contiene.

Análisis: Para la solución de este problema se requiere que el usuario ingrese el número inicial y final; luego, que el sistema procese y devuelva la cantidad de números pares que contiene el rango.

Entrada

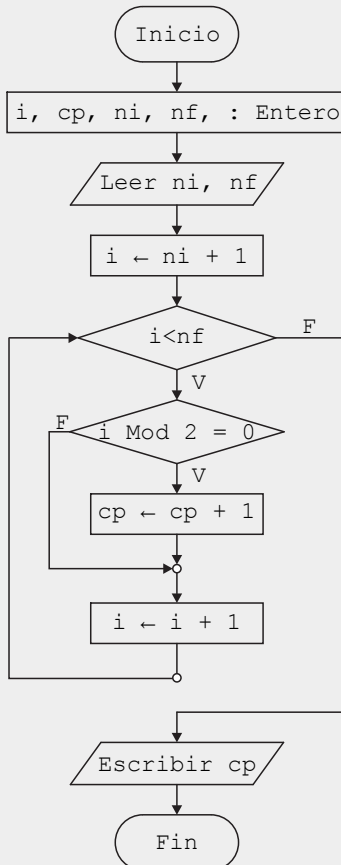
- Número inicial (ni)
- Número final (nf)

Salida

- Cantidad de pares (cp)

Diseño:
Interfaz de usuario

The screenshot shows a web browser window titled 'Problema 43 - Windows Internet Explorer'. The address bar shows 'http://localhost/Cap05/Problema43.php'. The browser menu includes 'Archivo', 'Edición', 'Ver', 'Favoritos', 'Herramientas', and 'Ayu...'. The page content is titled 'Problema 43' and contains three input fields: 'Num. Inicial' with the value '3', 'Num. Final' with the value '13', and 'Cant. Pares' with the value '5'. Below these fields is a 'Calcular' button.

Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio**

//Variables

`i, cp, ni, nf : Entero`

//Entrada

Leer ni, nf

//Proceso

`i ← ni + 1`

Mientras `i < nf`

 Si `i Mod 2 = 0` Entonces

`cp ← cp + 1`

 Fin Si

`i ← i + 1`

Fin Mientras

//Salida

Escribir cp

Fin

Codificación:

```

<?php
//Variables
$i = 0;$ni = 0;$nf = 0; $cp = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $ni = (int)$_POST["txtni"];
    $nf = (int)$_POST["txtnf"];

    //Proceso
    $i = $ni + 1;
    while($i < $nf){
        if($i % 2 == 0){
            $cp += 1;
        }
        $i++;
    }
}
?>
<html>
<head>
<title>Problema 43</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema43.php">
    <table width="288" border="0">
        <tr>
            <td colspan="2"><strong>Problema 43</strong> </td>
        </tr>
        <tr>
            <td width="87">Num. Inicial </td>
            <td width="144">
                <input name="txtni" type="text" id="txtni" value="<?=$ni?>"
            </td>
        </tr>
        <tr>
            <td>Num. Final </td>
            <td><input name="txtnf" type="text" id="txtnf" value="<?=$nf?>"
        </td>
        </tr>

        <tr>
            <td>Cant. Pares </td>
            <td><input name="txtcp" type="text" class="TextoFondo" id="txtcp"
value="<?=$cp?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 44

Enunciado: Obtener la cantidad de los primeros N números múltiplos de 5.

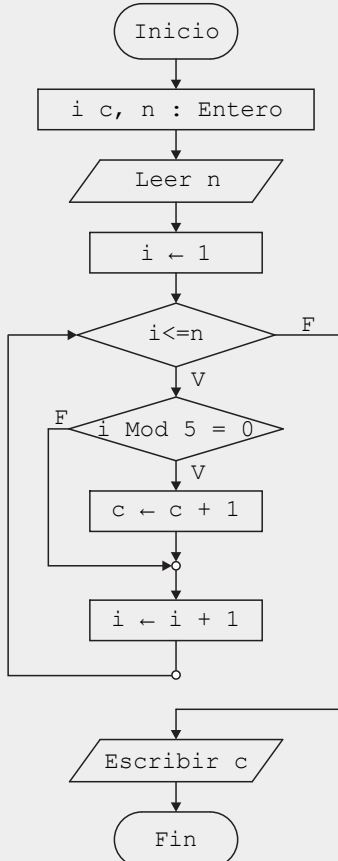
Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema devuelva la cantidad de números múltiplos de 5.

Entrada

- Número (n)

Salida

- Cantidad (c)

Diseño:**Interfaz de usuario****Algoritmo****Diagrama de flujo****Pseudocódigo****Inicio****//Variables**

i, c, n : Entero

//Entrada

Leer n

//Proceso

i ← 1

Mientras i <= n

 Si i Mod 5 = 0 Entonces

 c ← c + 1

 Fin Si

 i ← i + 1

Fin Mientras

//Salida

Escribir c

Fin

Codificación:

```

<?php
//Variables
$i = 0;$n = 0;$c = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $i = 1;
    while($i <= $n){
        if($i % 5 == 0){
            $c += 1;
        }
        $i++;
    }
}
?>
<html>
<head>
<title>Problema 44</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema44.php">
    <table width="241" border="0">
        <tr>
            <td colspan="2"><strong>Problema 44</strong> </td>
        </tr>
        <tr>
            <td width="73">Numero</td>
            <td width="146">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Cantidad</td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 45

Enunciado: Dado un número, determinar cuántos dígitos tiene.

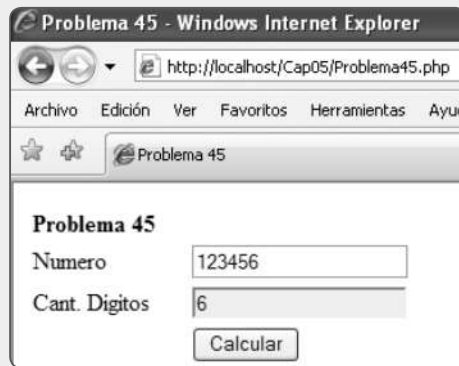
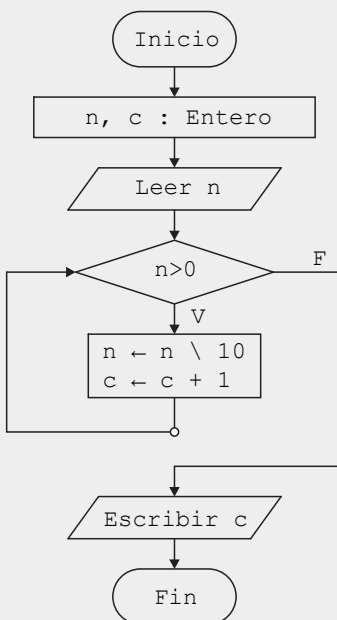
Análisis: Para la solución de este problema se requiere que el usuario ingrese un número entero; luego, que el sistema verifique y determine la cantidad de dígitos que contiene.

Entrada

- Número (n)

Salida

- Cantidad de dígitos (c)

Diseño:**Interfaz de usuario****Diagrama de flujo****Algoritmo****Pseudocódigo****Inicio**

//Variables

n, c : Entero

//Entrada

Leer n

//Proceso

Mientras n > 0

 n ← n \ 10

 c ← c + 1

Fin Mientras

//Salida

Escribir c

Fin

Codificación:

```

<?php
//Variables
$n = 0; $c = 0; $nn = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $nn = $n;
    while($n > 0) {
        $n = (int)($n / 10);
        $c += 1;
    }
    $n = $nn;
}
?>
<html>
<head>
<title>Problema 45</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema45.php">
    <table width="288" border="0">
        <tr>
            <td colspan="2"><strong>Problema 45</strong> </td>
        </tr>
        <tr>
            <td width="87">Numero</td>
            <td width="144">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Cant. Digitos </td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;  </td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 46

Enunciado: Dado un número, determinar la cantidad de dígitos pares que contiene.

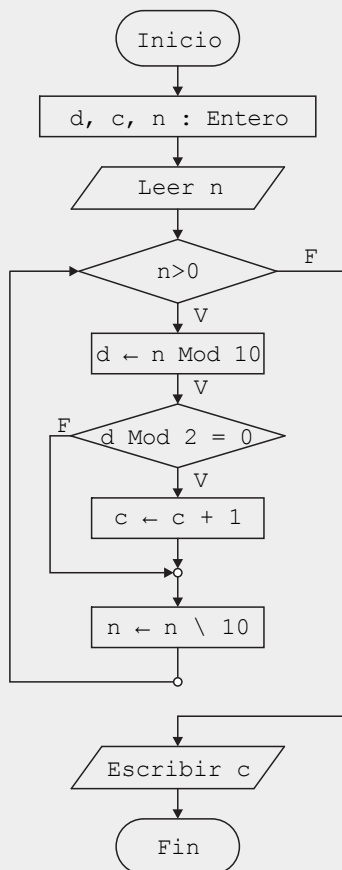
Análisis: Para la solución de este problema se requiere que el usuario ingrese un número entero; luego, que el sistema verifique y devuelva la cantidad de dígitos enteros que contiene el número.

Entrada

- Números (n)

Salida

- Cantidad de dígitos pares (c)

Diseño:**Interfaz de usuario****Algoritmo****Diagrama de flujo****Pseudocódigo****Inicio****//Variables**

d, c, n : Entero

//Entrada

Leer n

//Proceso

Mientras n > 0

 d ← n Mod 10

 Si d Mod 2 = 0

Entonces

 c ← c + 1

 Fin Si

 n ← n \ 10

Fin Mientras

//Salida

Escribir c

Fin

Codificación:

```

<?php

//Variables
$i = 0; $d = 0; $c = 0; $n = 0; $nn = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $nn = $n;
    while($n > 0){
        $d = $n % 10;
        if($d % 2 == 0) {
            $c += 1;
        }
        $n = (int)($n / 10);
    }
    $n = $nn;
}
?>
<html>
<head>
<title>Problema 46</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema46.php">
    <table width="288" border="0">
        <tr>
            <td colspan="2"><strong>Problema 46</strong> </td>
        </tr>
        <tr>
            <td width="127">Numero</td>
            <td width="151">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td><strong>Cant. Digitos Pares</strong> </td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 47

Enunciado: Dado un número, devolver el dígito mayor.

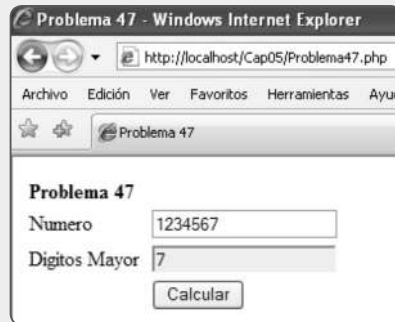
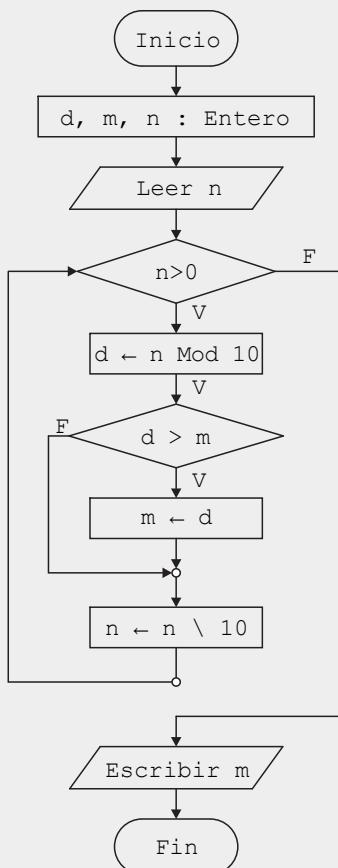
Análisis: Para la solución de este problema, se requiere que el usuario ingrese un número entero; luego, que el sistema verifique y devuelva el dígito mayor.

Entrada

- Número entero (n)

Salida

- Dígito mayor (m)

Diseño:**Interfaz de usuario****Algoritmo****Diagrama de flujo****Pseudocódigo****Inicio****//Variables**

d, m, n : Entero

//Entrada

Leer n

//Proceso

```

Mientras n > 0
    d ← n Mod 10
    Si d > m Entonces
        m ← d
    Fin Si
    n ← n \ 10
Fin Mientras
  
```

//Salida

Escribir m

Fin

Codificación:

```

<?php
//Variables
$d = 0; $m = 0; $n = 0; $nn = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $nn = $n;
    while($n > 0){
        $d = $n % 10;
        if($d > $m){
            $m = $d;
        }
        $n = (int)($n / 10);
    }
    $n = $nn;
}
?>
<html>
<head>
<title>Problema 47</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema47.php">
    <table width="276" border="0">
        <tr>
            <td colspan="2"><strong>Problema 47</strong> </td>
        </tr>
        <tr>
            <td width="100">Numero</td>
            <td width="178">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Digitos Mayor </td>
            <td><input name="txtm" type="text" class="TextoFondo" id="txtm"
value="<?=$m?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 48

Enunciado: Dados 2 números, diga si son amigos. Recuerde que dos números son amigos si la suma de sus divisores de uno de ellos es igual al otro y viceversa, por ejemplo, 220 y 284 son amigos:

Divisores de 220 son:

$$1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = \mathbf{284}$$

Divisores de 284 son:

$$1 + 2 + 4 + 71 + 142 = \mathbf{220}$$

Análisis: Para la solución de este problema se requiere que el usuario ingrese dos números (n_1 y n_2); luego, que el sistema verifique y devuelva si son o no número amigos.

Entrada

- Números (n_1 , n_2)

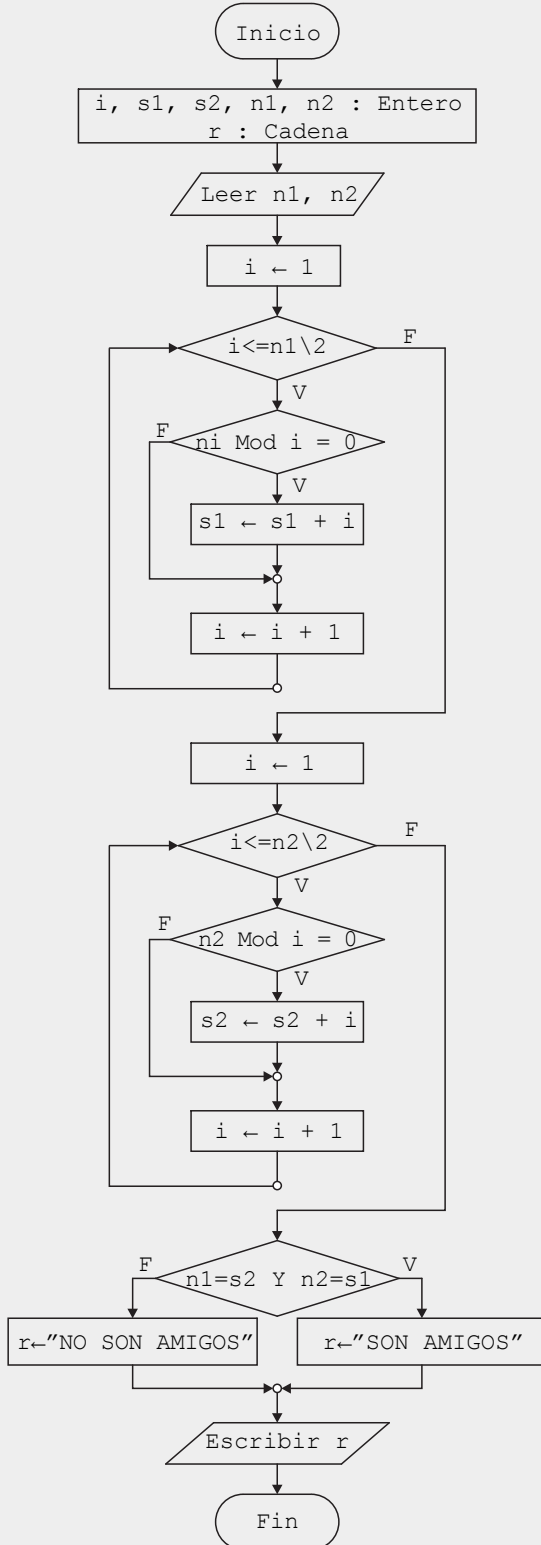
Salida

- Respuesta (r)
 - SON AMIGOS
 - NO SON AMIGOS

Diseño:**Interfaz de usuario**

The screenshot shows a web browser window titled "Problema 48 - Windows Internet Explorer". The address bar shows the URL "http://localhost/Cap05/Problema48.php". The browser has a menu bar with "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". Below the menu bar is a toolbar with a star icon and a plus icon. The main content area displays the title "Problema 48" in bold. Below the title are three input fields: "Numero 1" with the value "220", "Numero 2" with the value "284", and "Resultado" with the value "SON AMIGOS". At the bottom of the form is a button labeled "Calcular".

Diagrama de flujo



Algoritmo

Pseudocódigo

Inicio

//Variables

```
i, s1, s2, n1, n2 : Entero
r : Cadena
```

//Entrada

```
Leer n1, n2
```

//Proceso

```
i ← 1
Mientras i ≤ n1/2
    Si n1 Mod i = 0 Entonces
        s1 ← s1 + i
    Fin Si
    i = i + 1
Fin Mientras
```

```
i ← 1
Mientras i ≤ n2/2
    Si n2 Mod i = 0 Entonces
        s2 ← s2 + i
    Fin Si
    i = i + 1
Fin Mientras
Si n1 = s2 Y n2 = n1 Entonces
    r ← "SON AMIGOS"
SiNo
    r ← "NO SON AMIGOS"
Fin Si
```

//Salida

```
Escribir r
```

Fin

Codificación:

```

<?php

//Variables
$i = 0; $n1 = 0; $n2 = 0; $s1 = 0; $s2 = 0;
$r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n1 = (int)$_POST["txtn1"];
    $n2 = (int)$_POST["txtn2"];

    //Proceso
    $i = 1;
    while($i <= $n1 / 2){
        if($n1 % $i == 0){
            $s1 += $i;
        }
        $i++;
    }

    $i = 1;
    while($i <= $n2 / 2){
        if($n2 % $i == 0){
            $s2 += $i;
        }
        $i++;
    }

    if($n1 == $s2 && $n2 == $s1)
        $r = "SON AMIGOS";
    else
        $r = "NO SON AMIGOS";
}
?>
<html>
<head>
<title>Problema 48</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema48.php">
    <table width="237" border="0">
        <tr>
            <td colspan="2"><strong>Problema 48</strong> </td>
        </tr>
        <tr>
            <td width="82">Numero 1 </td>
            <td width="144">
                <input name="txtn1" type="text" id="txtn1" value="<?=$n1?>"
            </td>
        </tr>
        <tr>
            <td>Numero 2 </td>

```

```

        <td><input name="txtn2" type="text" id="txtn2" value="<?=$n2?>"
/></td>
    </tr>

    <tr>
        <td>Resultado</td>
        <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 49

Enunciado: Dado un número, devuelva el inverso del número.

Análisis: Para la solución de este problema se requiere que el usuario ingrese el número; luego, que el sistema procese y devuelva el inverso del número.

Entrada

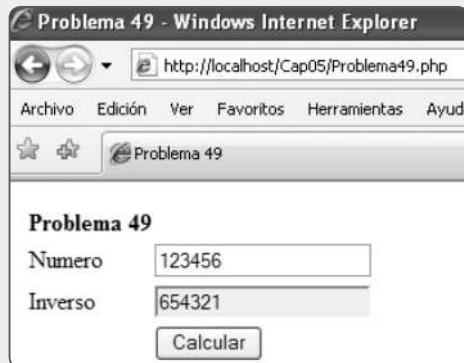
- Número (n)

Salida

- Número inverso (i)

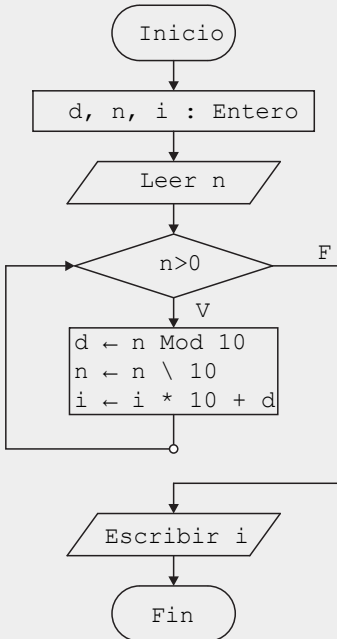
Diseño:

Interfaz de usuario



Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

d, n, i : Entero

//Entrada

Leer n

//Proceso

Mientras n > 0

 d ← n Mod 10

 n ← n \ 10

 i ← i * 10 + d

Fin Mientras

//Salida

Escribir i

Fin

Codificación:

```

<?php
//Variables
$d = 0; $n = 0; $i = 0; $nn = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $nn = $n;
    while($n > 0) {
        $d = $n % 10;
        $n = (int)($n / 10);
        $i = $i * 10 + $d;
    }
    $n = $nn;
}
?>
<html>
<head>
<title>Problema 49</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
  
```

```

<form method="post" action="Problema49.php">
  <table width="237" border="0">
    <tr>
      <td colspan="2"><strong>Problema 49</strong> </td>
    </tr>
    <tr>
      <td width="82">Numero </td>
      <td width="144">
        <input name="txtn" type="text" id="txtn" value="<?=$n?" />
      </td>
    </tr>
    <tr>
      <td>Inverso </td>
      <td><input name="txti" type="text" class="TextoFondo" id="txti"
value="<?=$i?" /></td>
    </tr>

    <tr>
      <td>&nbsp;</td>
      <td>
        <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
      </td>
    </tr>
  </table>
</form>
</body>
</html>

```

Problema n.º 50

Enunciado: Crear un algoritmo que indique si un número es cubo perfecto (anstrong) o no; se dice que un número es cubo perfecto si al sumar los cubos de sus dígitos dan el mismo número. Por ejemplo: 153, los cubos de sus dígitos $1^3 + 5^3 + 3^3 = 153$. Por lo tanto, el número 153 es cubo perfecto.

Análisis: Para la solución de este problema se requiere que el usuario ingrese el número; luego, que el sistema procese y determine si es o no un cubo perfecto.

Entrada

- Número (n)

Salida

- Respuesta (r)
 - CUBO PERFECTO
 - NO ES CUBO PERFECTO

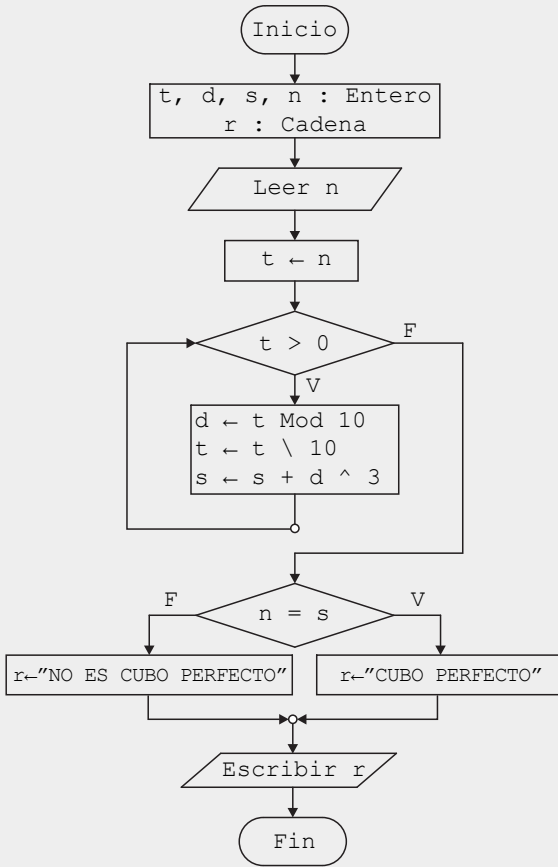
Diseño:

Interfaz de usuario



Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables
 t, d, s, n : Entero
 r : Cadena

//Entrada

Leer n

//Proceso

t ← n
Mientras t > 0
 d ← t Mod 10
 t ← t \ 10
 s ← s + d ^ 3

Fin Mientras

Si n = s **Entonces**
 r ← "CUBO PERFECTO"
SiNo
 r ← "NO ES CUBO PERFECTO"
Fin Si

//Salida

Escribir r

Fin

Codificación:

```

<?php

//Variables
$t = 0; $d = 0; $s = 0;  $n = 0;
$r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $t = $n;
    while($t > 0){
        $d = $t % 10;
        $t = (int)($t / 10);
        $s = (int)($s + pow((double)$d, 3.0));
    }

    if($n == $s)
        $r = "CUBO PERFECTO";
    else
        $r = "NO ES CUBO PERFECTO";
}
  
```

```

?>
<html>
<head>
<title>Problema 50</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema50.php">
    <table width="237" border="0">
        <tr>
            <td colspan="2"><strong>Problema 50</strong> </td>
        </tr>
        <tr>
            <td width="82">Numero </td>
            <td width="144">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>"
            </td>
        </tr>
        <tr>
            <td>Resultado</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 51

Enunciado: Obtenga el cociente y el residuo de una división mediante restas sucesivas; por ejemplo, si el dividendo es 3989 y el divisor es 1247, entonces:

$$3989 - 1247 = 2742 \quad R(1)$$

$$2742 - 1247 = 1495 \quad R(2)$$

$$1495 - 1247 = 248 \quad R(3)$$

Ya no se puede seguir restando, pues 248 es menor a 1247; entonces, el cociente es el número de veces restado (3) y el residuo es el último número obtenido (248).

Análisis: Para la solución de este problema, se requiere que el usuario ingrese la temperatura; luego, que el sistema verifique y determine el clima.

Entrada

- Numerador (n)
- Denominador (d)

Salida

- Cociente (c)
- Residuo (r)

Diseño: Interfaz de usuario

Problema 51

Numerador:

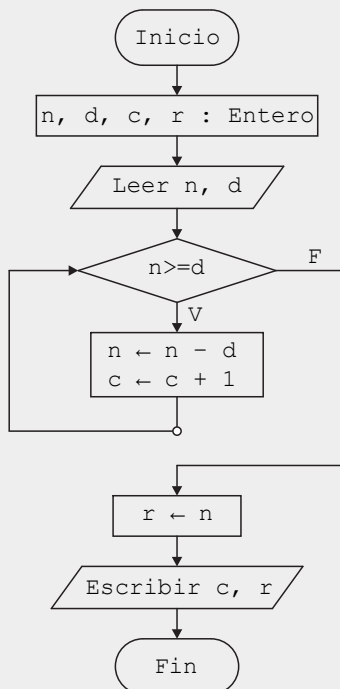
Denominador:

Cociente:

Residuo:

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

$n, d, c, r : \text{Entero}$

//Entrada

Leer n, d

//Proceso

Mientras $n \geq d$

$n \leftarrow n - d$

$c \leftarrow c + 1$

Fin Mientras

$r \leftarrow n$

//Salida

Escribir c, r

Fin

Codificación:

```

<?php

//Variables
$n = 0; $d = 0; $c = 0; $r = 0; $nn = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];
    $d = (int)$_POST["txtd"];

    //Proceso
    $nn = $n;
    while($n >= $d){
        $n -= $d;
        $c++;
    }
    $r = $n;
    $n = $nn;
}
?>
<html>
<head>
<title>Problema 51</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema51.php">
    <table width="245" border="0">
        <tr>
            <td colspan="2"><strong>Problema 51</strong> </td>
        </tr>
        <tr>
            <td width="89">Numerador</td>
            <td width="146">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Denominador</td>
            <td><input name="txtd" type="text" id="txtd" value="<?=$d?>" />
        </td>
        </tr>
    </table>
</form>

```

```

<tr>
    <td>Cociente</td>
    <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" /></td>
</tr>
<tr>
    <td>Residuo</td>
    <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>
        <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
    </td>
</tr>
</table>
</form>
</body>
</html>

```

Problema n.º 52

Enunciado: Determine si un número es capicúa o no. Se dice que un número capicúa es aquel cuyas cifras, al ser invertidas, dan el mismo número. Por ejemplo, 12 321 invertido es 12 321, entonces es un número capicúa.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema verifique y determine si es o no capicúa.

Entrada

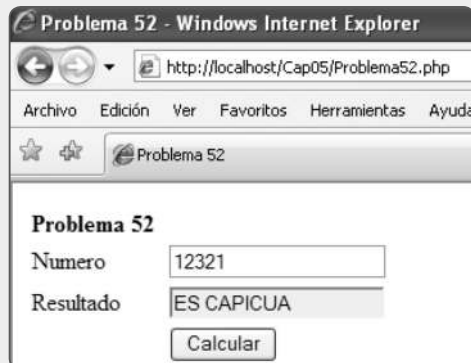
- Número (n)

Salida

- Respuesta (r)
 - ES CAPICUA
 - NO ES CAPICUA

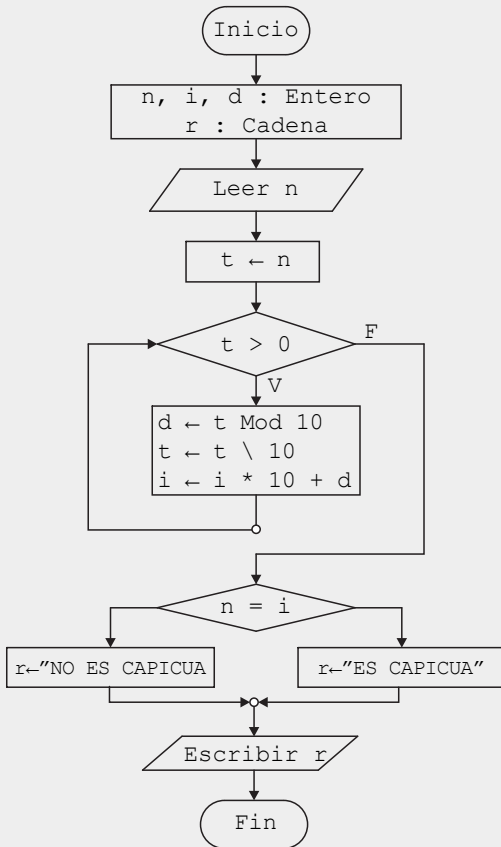
Diseño:

Interfaz de usuario



Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

n, i, d : Entero
r : Cadena

//Entrada

Leer n

//Proceso

t ← n
Mientras t > 0
 d ← t Mod 10
 t ← t \ 10
 i ← i * 10 + d

Fin Mientras

Si n = i Entonces
 r ← "ES CAPICUA"

SiNo

 r ← "NO ES CAPICUA"

Fin Si

//Salida

Escribir r

Fin

Codificación:

```

<?php
//Variables
$n = 0; $i = 0; $d = 0; $t = 0;
$r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $t = $n;
    while($t > 0){
        $d = $t % 10;
        $t = (int)($t / 10);
        $i = $i * 10 + $d;
    }

    if($n == $i)
        $r = "ES CAPICUA";
    else
        $r = "NO ES CAPICUA";
}
  
```

```

?>
<html>
<head>
<title>Problema 52</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema52.php">
    <table width="245" border="0">
        <tr>
            <td colspan="2"><strong>Problema 52</strong> </td>
        </tr>
        <tr>
            <td width="89">Numero</td>
            <td width="146">
                <input name="txtn" type="text" id="txtn" value="<?=$n?" />
            </td>
        </tr>
        <tr>
            <td><strong>Resultado</strong>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 53

Enunciado: Dado un número, determine si es primo, recuerde que un número primo es aquel que solo es divisible por 1 y por sí mismo.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema determine si es primo o no.

Entrada

- Número (n)

Salida

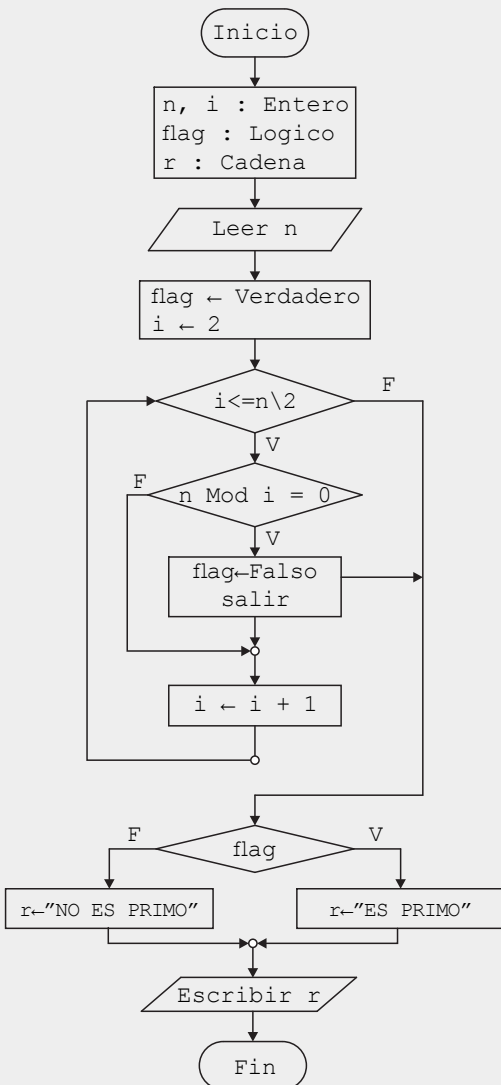
- Respuesta (r)
 - ES PRIMO
 - NO ES PRIMO

Diseño:
Interfaz de usuario

The screenshot shows a web browser window titled 'Problema 54 - Windows Internet Explorer'. The address bar shows 'http://localhost/Cap05/Problema54.php'. The browser has a menu bar with 'Archivo', 'Edición', 'Ver', 'Favoritos', 'Herramientas', and 'Ayuda'. Below the menu is a toolbar with a star icon and the text 'Problema 54'. The main content area is titled 'Problema 54' and contains three input fields: 'Numero' with the value '12345', 'Base' with the value '6', and 'Resultado' with the text 'BASE CORRECTA'. Below these fields is a button labeled 'Calcular'.

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

n, i : Entero
flag : Logico
r : Cadena

//Entrada

Leer n

//Proceso

flag ← Verdadero
i ← 2

Mientras i <= n/2

 Si n Mod i = 0

 flag ← Falso

 Salir

 Fin Si

 i ← i + 1

Fin Mientras

Si flag Entonces

 r ← "ES PRIMO"

SiNo

 r ← "NO ES PRIMO"

Fin Si

//Salida

Escribir r

Fin

Codificación:

```
<?php

//Variables
$n = 0; $i = 0;
$flag = false;
$r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txttn"];

    //Proceso
    $flag = true;
    $i = 2;
    while($i <= $n / 2){
        if($n % $i == 0){
            $flag = false;
            break;
        }
        $i++;
    }

    if($flag)
        $r = "ES PRIMO";
    else
        $r = "NO ES PRIMO";
}
?>
<html>
<head>
<title>Problema 53</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema53.php">
    <table width="245" border="0">
        <tr>
            <td colspan="2"><strong>Problema 53</strong> </td>
        </tr>
        <tr>
            <td width="89">Numero</td>
            <td width="146">
                <input name="txttn" type="text" id="txttn" value="<?=$n?>" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>
```

```

        <tr>
            <td>Respuesta</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 54

Enunciado: Dado un número y su base, determine si el número pertenece a la base ingresada; recuerde que un número pertenece a una base si sus dígitos son menores a su base.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema verifique y determine si pertenece a la base o no.

Entrada

- Número (n)
- Base (b)

Salida

- Respuesta (r)
 - BASE CORRECTA
 - BASE INCORRECTA

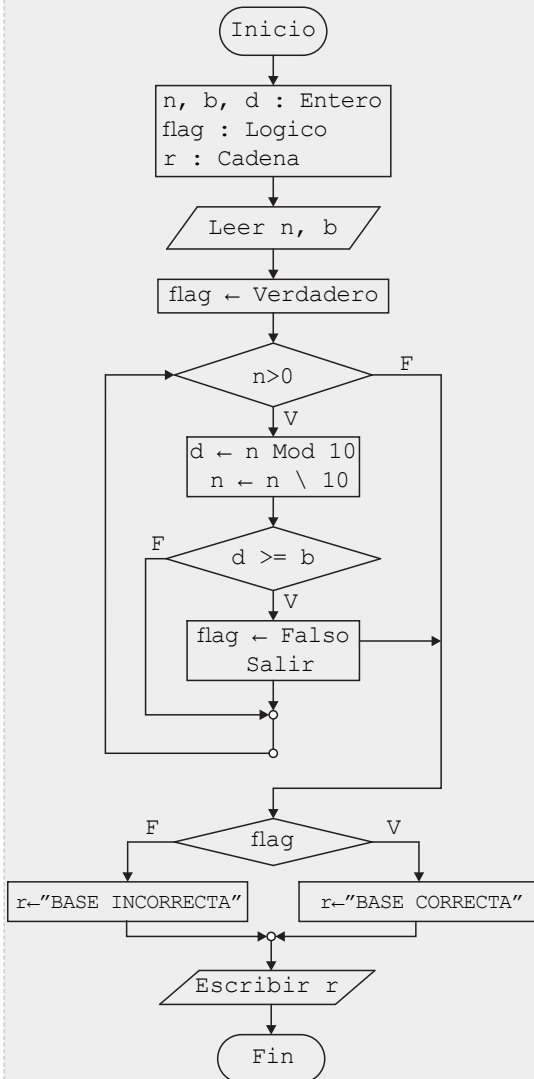
Diseño:

Interfaz de usuario

The screenshot shows a web browser window titled "Problema 54 - Windows Internet Explorer". The address bar displays the URL "http://localhost/Cap05/Problema54.php". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The page content is titled "Problema 54" and contains a form with three input fields: "Numero" (containing "12345"), "Base" (containing "6"), and "Resultado" (containing "BASE CORRECTA"). Below these fields is a "Calcular" button.

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

```

n, b, d : Entero
flag : Logico
r : Cadena

```

//Entrada

```

Leer n, b

```

//Proceso

```

flag ← Verdadero
Mientras n > 0
    d ← n Mod 10
    n ← n \ 10
    Si d >= b Entonces
        flag ← Falso
        Salir
    Fin Si
Fin Mientras

Si flag Entonces
    r ← "BASE CORRECTA"
SiNo
    r ← "BASE INCORRECTA"
Fin Si

```

//Salida

```

Escribir r

```

Fin

Codificación:

```

<?php

//Variables
$n = 0; $b = 0; $d = 0; $nn = 0;
$flag = false;
$r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];
    $b = (int)$_POST["txtb"];

    //Proceso
    $nn = $n;
    $flag = true;
    while($n > 0) {
        $d = $n % 10;
        $n = (int)($n / 10);
        if($d >= $b){
            $flag = false;
            break;
        }
    }
    $n = $nn;
    if($flag)
        $r = "BASE CORRECTA";
    else
        $r = "BASE INCORRECTA";
}
?>
<html>
<head>
<title>Problema 54</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema54.php">
    <table width="245" border="0">
        <tr>
            <td colspan="2"><strong>Problema 54</strong> </td>
        </tr>
        <tr>
            <td width="89">Numero</td>
            <td width="146">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>

```

```

        <td>Base</td>
        <td><input name="txtb" type="text" id="txtb" value="<?=$b?>"
/></td>
    </tr>

    <tr>
        <td>Resultado</td>
        <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 55

Enunciado: Dado un número entero en base 10, convertir el número a otra base menor que 10.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese el número entero y la base a convertir; luego, que el sistema devuelva el número convertido a su nueva base.

Entrada

- Número (n)
- Base (b)

Salida

- Número convertido (r)

Diseño:

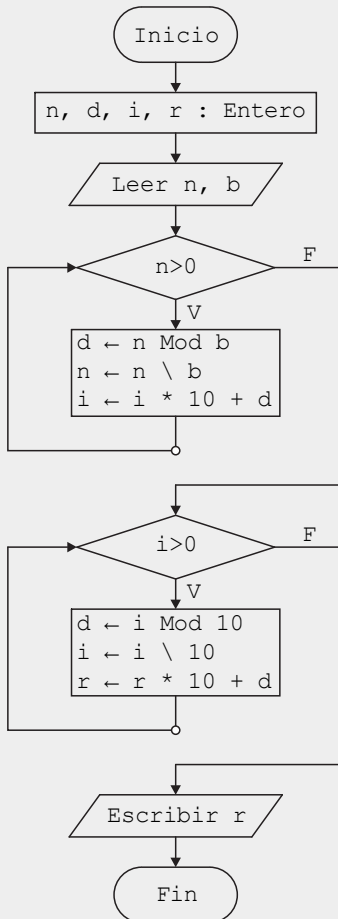
Interfaz de usuario

The screenshot shows a web browser window titled "Problema 55 - Windows Internet Explorer". The address bar shows "http://localhost/Cap05/Problema55.php". The browser menu includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The page content is titled "Problema 55" and contains the following form elements:

- A label "Numero Base 10" followed by a text input field containing the value "123".
- A label "Convertir a Base" followed by a text input field containing the value "2".
- A label "Resultado" followed by a text input field containing the value "1111011".
- A "Calcular" button located below the "Resultado" field.

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

n, d, i, r : Entero

//Entrada

Leer n, b

//Proceso

Mientras n > 0

d ← n Mod b

n = n \ 10

i = i * 10 + d

Fin Mientras

Mientras i > 0

d ← i Mod 10

i = i \ 10

r = r * 10 + d

Fin Mientras

//Salida

Escribir r

Fin

Codificación:

```

<?php

//Variables
$n = 0; $b = 0; $d = 0; $i = 0; $r = 0; $nn = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txttn"];
    $b = (int)$_POST["txtb"];

    //Proceso
    $nn = $n;
    while($n > 0) {
        $d = $n % $b;
        $n = (int)($n / $b);
        $i = $i * 10 + $d;
    }
}
  
```

```

        while($i > 0){
            $d = $i % 10;
            $i = (int)($i / 10);
            $r = $r * 10 + $d;
        }
        $n = $nn;
    }
    ?>
<html>
<head>
<title>Problema 55</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema55.php">
    <table width="279" border="0">
        <tr>
            <td colspan="2"><strong>Problema 55</strong> </td>
        </tr>
        <tr>
            <td width="115">Numero Base 10 </td>
            <td width="154">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Convertir a Base</td>
            <td><input name="txtb" type="text" id="txtb" value="<?=$b?>" /></td>
        </tr>
        <tr>
            <td>Resultado</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr" value="<?=$r?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular" value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

5.8 Problemas propuestos

Los siguientes problemas le servirán para medir su aprendizaje, es importante que los resuelva.

Propuesto n.º 31

Enunciado: Obtener el factorial de un número; recuerde que el factorial de un número es el producto de $1 \times 2 \times 3 \times \dots \times N$.

Propuesto n.º 32

Enunciado: Dado un rango de números enteros, obtener la cantidad de números pares e impares que contiene el rango, sin considerar los múltiplos de 5.

Propuesto n.º 33

Enunciado: Calcular la suma y el producto de los N primeros números naturales múltiplos de 3.

Propuesto n.º 34

Enunciado: Dado un número, determinar cuántos dígitos «0» contiene.

Propuesto n.º 35

Enunciado: Se requiere saber si existe un determinado dígito en un número dado.

Propuesto n.º 36

Enunciado: Dado un número, determinar cuál es el porcentaje de números pares, impares y neutros (0).

Propuesto n.º 37

Enunciado: Dado un rango de números, determine cuántos números primos contiene.

Propuesto n.º 38

Enunciado: Dado un rango de números, determine cuántos números capicúa hay.

Propuesto n.º 39

Enunciado: Dado 2 números, obtener el MCD (máximo común divisor); utilice el método EUCLIDES (divisiones sucesivas).

Propuesto n.º 40

Enunciado: Dado 2 números obtener el MCD (máximo común divisor), utilice el método factorización simultánea.

Recuerde: El máximo común divisor es el divisor mayor común de todos ellos.

Estructura repetitiva «Para»

6.1 Introducción

Cada vez que requiere repetir un proceso una determinada cantidad de veces, deberá usar la estructura repetitiva «Para» (for), que permitirá realizar en forma simple este trabajo.

Esta estructura usa una variable «contador», donde se establece el valor inicial (**vi**), valor final (**vf**) y el valor de incremento (**inc**), que determina las veces a repetir la instrucción.

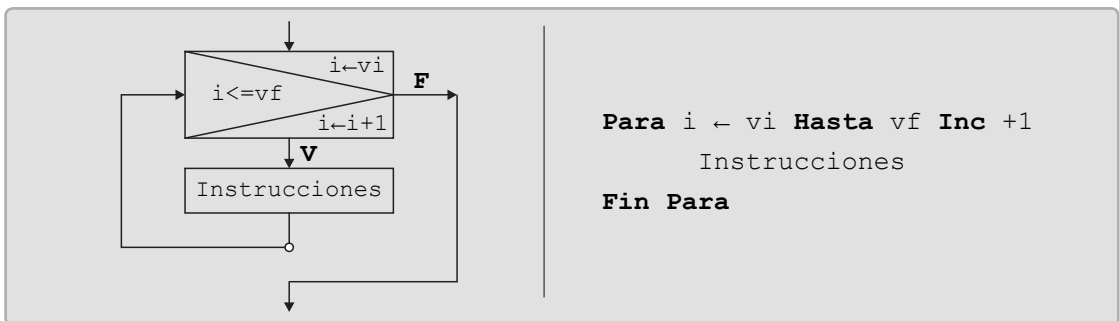
6.2 Estructura repetitiva «Para»

Permite repetir una o más instrucciones una cantidad de veces.

i Es nuestra variable contador, donde establecemos el valor inicial.

vf Representa el valor final de la variable contador.

+1 Valor de incremento.

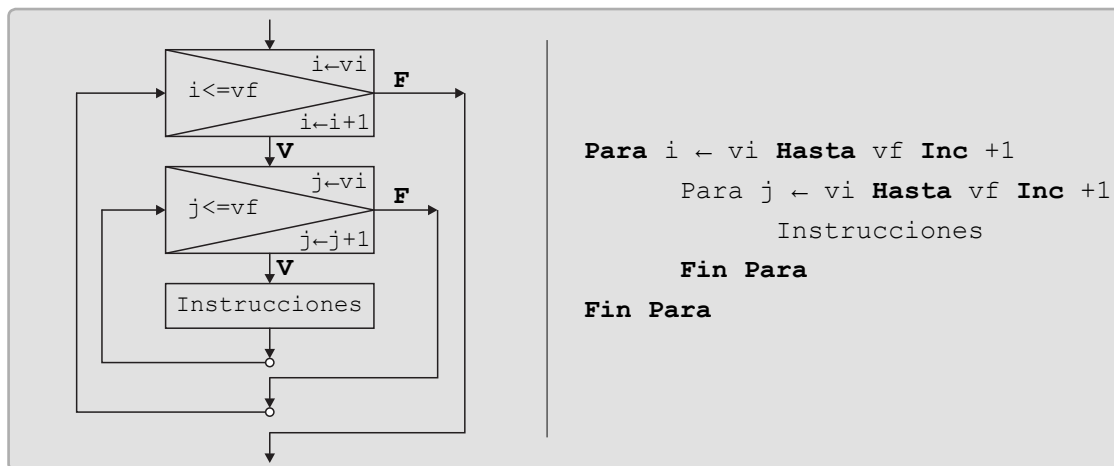


Sintaxis PHP

```
for ($i=vi;$i<=vf;$i++) {
    <instrucciones>;
}
```

6.3 Estructura repetitiva «Para» anidada

Dentro de la estructura repetitiva es posible colocar una o más estructuras repetitivas, así como otras estructuras.



Sintaxis PHP

```

for ($i=vi;$i<=vf;$i++) {
    for ($j=vi;$j<=vf;$j++) {
        <instrucciones>;
    }
}
  
```

Problema n.º 56

Enunciado: Obtener la suma de los primeros N números naturales positivos.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema realice el proceso para devolver la suma de los N primeros números.

Entrada

- Número (n)

Salida

- Suma (s)

Diseño:

Interfaz de usuario

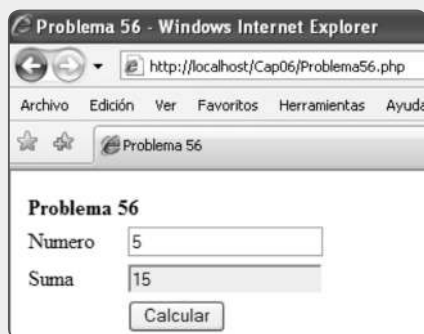
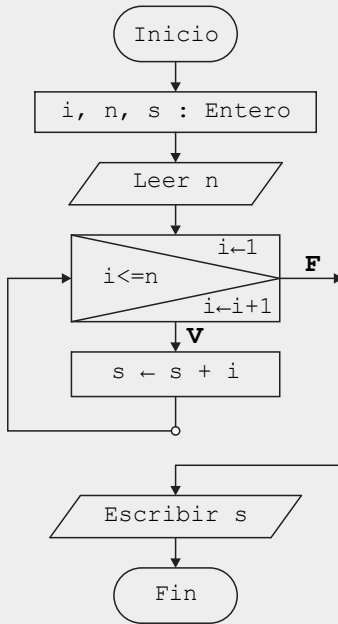


Diagrama de flujo**Algoritmo****Inicio****//Variables**`i, n, s : Entero`**//Entrada**`Leer n`**//Proceso**`Para i←1 Hasta n Inc 1``s ← s + i``Fin Para`**//Salida**`Escribir s`**Fin****Pseudocódigo****Codificación:**

```

<?php
//Variables
$i = 0; $n = 0; $s = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    for ($i = 1; $i<=$n; $i++)
        $s = $s + $i;
}
?>
<html>
<head>
<title>Problema 56</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema56.php">
    <table width="231" border="0">
        <tr>

```

```

        <td colspan="2"><strong>Problema 56</strong> </td>
    </tr>
    <tr>
        <td width="69">Numero</td>
        <td width="144">
            <input name="txtn" type="text" id="txtn" value="<?=$n?" />
        </td>
    </tr>
    <tr>
        <td>Suma</td>
        <td><input name="txts" type="text" class="TextoFondo" id="txts"
value="<?=$s?" /></td>
    </tr>

    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 57

Enunciado: Dado un rango de números enteros, obtener la cantidad de números enteros que contiene.

Análisis: Para la solución de este problema se requiere que el usuario ingrese el número inicial y final; luego, que el sistema procese y devuelva la cantidad de números enteros que contiene el rango.

Entrada

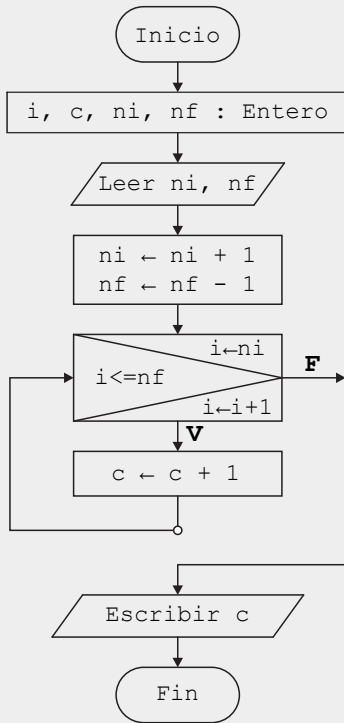
- Número Inicial (ni)
- Número Final (nf)

Salida

- Cantidad (c)

Diseño:

Interfaz de usuario

Diagrama de flujo**Algoritmo****Pseudocódigo****Inicio****//Variables**`i, c, ni, nf : Entero`**//Entrada**`Leer ni, nf`**//Proceso**`ni ← ni + 1``nf ← nf - 1``Para i←ni Hasta nf Inc 1``c ← c + 1``Fin Para`**//Salida**`Escribir c`**Fin****Codificación:**

```

<?php
//Variables
$i = 0; $ni = 0; $nf = 0; $c = 0;
$ni = 0; $nf = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $ni = (int)$_POST["txtni"];
    $nf = (int)$_POST["txtnf"];

    //Proceso
    $ni = $ni;
    $nf = $nf;
    $ni = $ni + 1;
    $nf = $nf - 1;

    for($i = $ni; $i <= $nf; $i++)
        $c += 1;

    $ni = $ni;
    $nf = $nf;
}
?>
<html>

```

```

<head>
<title>Problema 57</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema57.php">
    <table width="274" border="0">
        <tr>
            <td colspan="2"><strong>Problema 57</strong> </td>
        </tr>
        <tr>
            <td width="108">Num. Inicial </td>
            <td width="156">
                <input name="txtni" type="text" id="txtni" value="<?=$ni?>"
            </td>
        </tr>
        <tr>
            <td>Num. Final </td>
            <td><input name="txtnf" type="text" id="txtnf" value="<?=$nf?>"
        </td>
        </tr>

        <tr>
            <td>Cantidad</td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 58

Enunciado: Dado un rango de números enteros, obtener la cantidad de números pares que contiene.

Análisis: Para la solución de este problema se requiere que el usuario ingrese el número inicial y final; luego, que el sistema procese y devuelva la cantidad números pares que contiene el rango.

Entrada

- Número inicial (ni)
- Número final (nf)

Salida

- Cantidad de pares (cp)

Diseño:
Interfaz de usuario

Problema 58

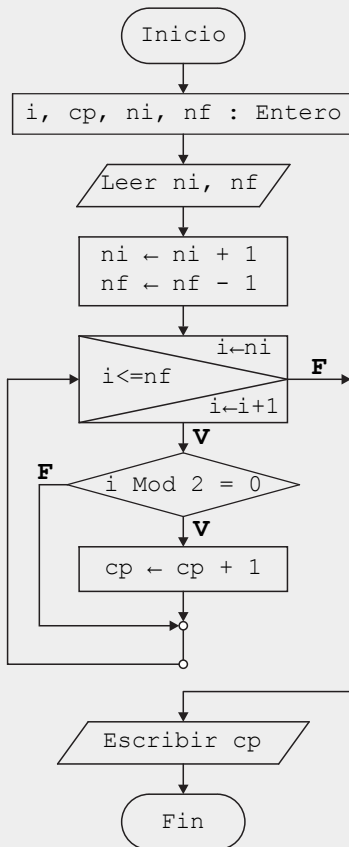
Num. Inicial

Num. Final

Cant. Pares

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

i, cp, ni, nf : Entero

//Entrada

Leer ni, nf

//Proceso

ni ← ni + 1

nf ← nf - 1

Para i ← ni Hasta nf Inc 1

 Si i Mod 2 = 0 Entonces

 cp ← cp + 1

 Fin Si

Fin Para

//Salida

Escribir cp

Fin

Codificación:

```

<?php
//Variables
$i = 0; $ni = 0; $nf = 0; $cp = 0;
$nii = 0; $nff = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $ni = (int)$_POST["txtni"];
    $nf = (int)$_POST["txtnf"];

    //Proceso
    $nii = $ni;
    $nff = $nf;
    $ni = $ni + 1;
    $nf = $nf - 1;
    for($i = $ni; $i <= $nf; $i++){
        if($i % 2 == 0)
            $cp += 1;
    }
    $ni = $nii;
    $nf = $nff;
}
?>
<html>
<head>
<title>Problema 58</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema58.php">
    <table width="274" border="0">
        <tr>
            <td colspan="2"><strong>Problema 58</strong> </td>
        </tr>
        <tr>
            <td width="108">Num. Inicial </td>
            <td width="156">
                <input name="txtni" type="text" id="txtni" value="<?=$ni?>" />
            </td>
        </tr>
        <tr>
            <td>Num. Final </td>
            <td><input name="txtnf" type="text" id="txtnf" value="<?=$nf?>" /></td>
        </tr>
        <tr>
            <td>Cant. Pares </td>
            <td><input name="txtcp" type="text" class="TextoFondo" id="txtcp" value="<?=$cp?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular" value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 59

Enunciado: Obtener la cantidad de los primeros N números múltiplos de 5.

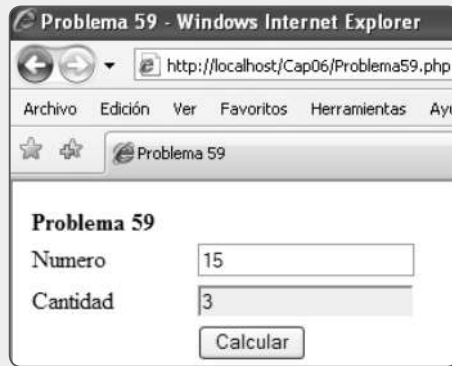
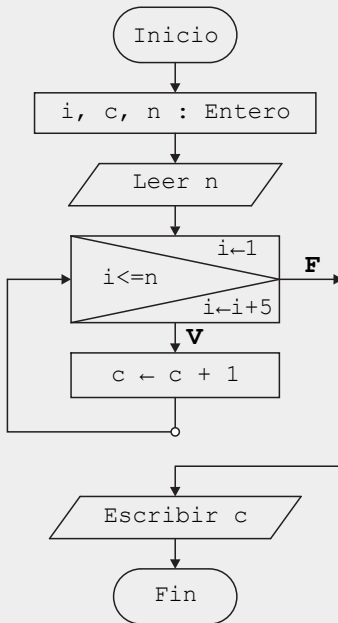
Análisis: Para la solución de este problema, se requiere que el usuario ingrese un número; luego, que el sistema devuelva la cantidad de números múltiplos de 5.

Entrada

- Número (n)

Salida

- Cantidad (c)

Diseño:**Interfaz de usuario****Diagrama de flujo****Algoritmo****Pseudocódigo****Inicio****//Variables**

i, c, n : Entero

//Entrada

Leer n

//Proceso

Para i←1 Hasta n Inc 5

 c ← c + 1

Fin Para

//Salida

Escribir c

Fin

Codificación:

```

<?php

//Variables
$i = 0; $n = 0; $c = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    for($i = 1; $i<=$n; $i+=5)
        $c += 1;
}
?>
<html>
<head>
<title>Problema 59</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema59.php">
    <table width="274" border="0">
        <tr>
            <td colspan="2"><strong>Problema 59</strong> </td>
        </tr>
        <tr>
            <td width="108">Numero</td>
            <td width="156">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Cantidad</td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 60

Enunciado: Obtener la suma de pares e impares de los primeros N números enteros positivos.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema devuelva la suma de pares e impares.

Entrada

- Número (n)

Salida

- Suma pares (sp)
- Suma impares (si)

Diseño: Interfaz de usuario

Problema 60 - Windows Internet Explorer

http://localhost/Cap06/Problema60.php

Archivo Edición Ver Favoritos Herramientas Ayud

Problema 60

Problema 60

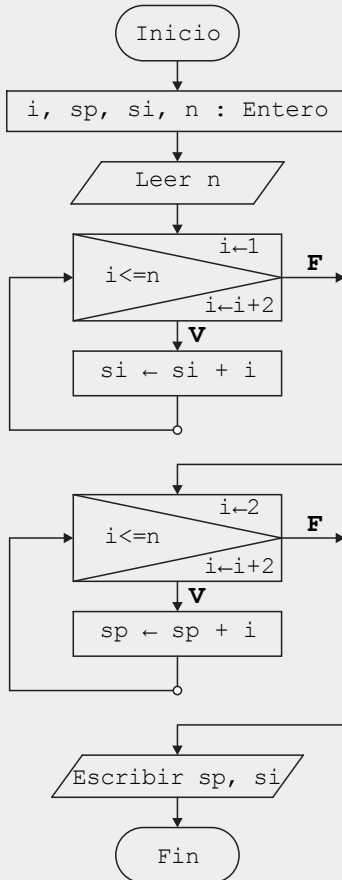
Numero

Suma Pares

Suma Impares

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

i, sp, si, n : Entero

//Entrada

Leer n

//Proceso

Para i←1 Hasta n Inc 2

 si ← si + i

Fin Para

Para i←2 Hasta n Inc 2

 sp ← sp + i

Fin Para

//Salida

Escribir sp, si

Fin

Codificación:

```

<?php
//Variables
$i = 0; $n = 0; $sp = 0; $si = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    for($i = 1; $i <= $n; $i += 2){
        $si += $i;
    }

    for($i = 2; $i <= $n; $i += 2){
        $sp += $i;
    }
}

?>
<html>
<head>
<title>Problema 60</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema60.php">
    <table width="274" border="0">
        <tr>
            <td colspan="2"><strong>Problema 60</strong> </td>
        </tr>
        <tr>
            <td width="108">Numero</td>
            <td width="156">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Suma Pares </td>
            <td><input name="txtsp" type="text" class="TextoFondo" id="txtsp"
value="<?=$sp?>" /></td>
        </tr>
        <tr>
            <td>Suma Impares </td>
            <td><input name="txtsi" type="text" class="TextoFondo" id="txtsi"
value="<?=$si?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 61

Enunciado: Hallar el cuadrado de un número usando la siguiente relación: $N^2 = 1 + 3 + 5 + \dots + 2N - 1$.

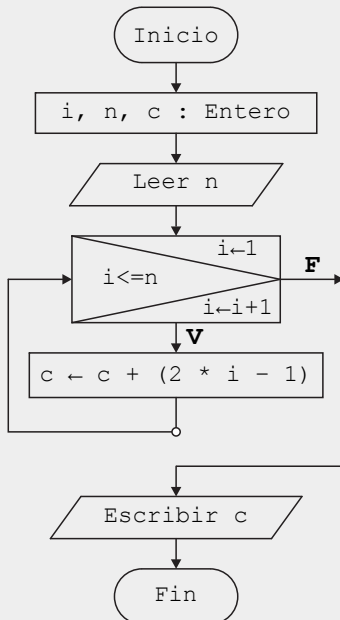
Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema devuelve el cuadrado del número.

Entrada

- Número (n)

Salida

- Cuadrado (c)

Diseño:**Interfaz de usuario****Diagrama de flujo****Algoritmo****Pseudocódigo****Inicio**

//Variables

i, n, c : Entero

//Entrada

Leer n

//Proceso

Para i ← 1 Hasta n Inc 1

c ← c + (2 * i - 1)

Fin Para

//Salida

Escribir c

Fin

Codificación:

```

<?php
//Variables
$i = 0; $n = 0; $c = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    for($i = 1; $i <= $n; $i++)
        $c = $c + (2 * $i - 1);
}
?>
<html>
<head>
<title>Problema 61</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema61.php">
    <table width="244" border="0">
        <tr>
            <td colspan="2"><strong>Problema 61</strong> </td>
        </tr>
        <tr>
            <td width="77">Numero</td>
            <td width="148">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td></td>
            <td>Cuadrado</td>
        </tr>
        <tr>
            <td colspan="2">
                <input name="txtc" type="text" class="TextoFondo" id="txtc"
                value="<?=$c?>" />
            </td>
        </tr>
        <tr>
            <td colspan="2">
                <input name="btnCalcular" type="submit" id="btnCalcular"
                value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 62

Enunciado: Crear el algoritmo que indique si un número es perfecto o no; se dice que un número es perfecto si la suma de sus divisores es igual al número. Por ejemplo, 6 tiene como divisores 1, 2 y 3; entonces $1 + 2 + 3 = 6$, el número 6 es perfecto. Si el número es 9, tiene como divisores 1 y 3; entonces $1 + 3 = 4$, por lo tanto no es perfecto.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema devuelva si el número es o no perfecto.

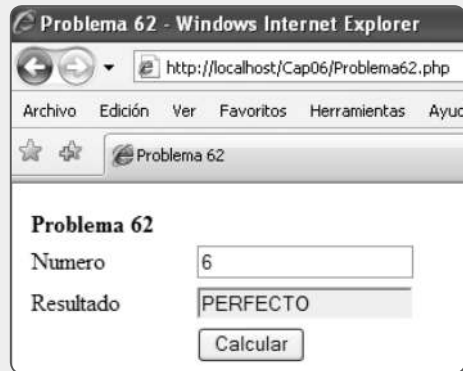
Entrada

- Número (n)

Salida

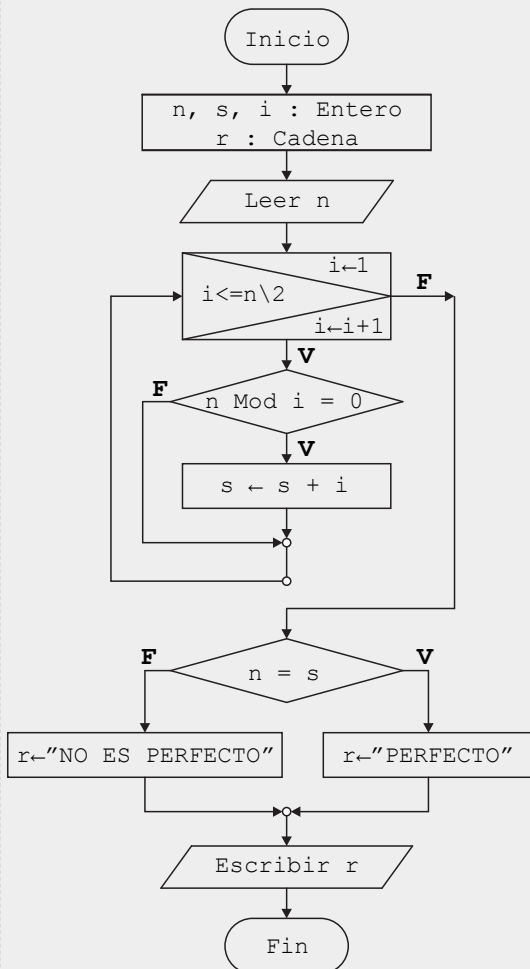
- Respuesta (r)

Diseño: Interfaz de usuario



Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

```
n, s, i : Entero
r : Cadena
```

//Entrada

```
Leer n
```

//Proceso

```
Para i ← 1 Hasta n \ 2 Inc 1
    Si n Mod i = 0 Entonces
        s ← s + i
    Fin Si
Fin Para
```

```
Si n = s Entonces
    r ← "PERFECTO"
```

```
SiNo
```

```
    r ← "NO ES PERFECTO"
```

```
Fin Si
```

//Salida

```
Escribir r
```

Fin

Codificación:

```

<?php

//Variables
$n = 0; $s = 0; $i = 0;
$r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    for($i = 1; $i <= $n / 2; $i++) {
        if($n % $i == 0)
            $s += $i;
    }

    if($n == $s)
        $r = "PERFECTO";
    else
        $r = "NO ES PERFECTO";
}
?>
<html>
<head>
<title>Problema 62</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema62.php">
    <table width="274" border="0">
        <tr>
            <td colspan="2"><strong>Problema 62</strong> </td>
        </tr>
        <tr>
            <td width="108">Numero</td>
            <td width="156">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Resultado</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>

```

```
</tr>

<tr>
  <td>&nbsp;</td>
  <td>
    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
  </td>
</tr>
</table>
</form>
</body>
</html>
```

Problema n.º 63

Enunciado: Dados 2 números, diga si son amigos o no; recuerde que dos números son amigos si la suma de los divisores de uno de ellos es igual al otro y viceversa, por ejemplo 220 y 284 son amigos:

Divisores de 220 son: $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$

Divisores de 284 son: $1 + 2 + 4 + 71 + 142 = 220$

Análisis: Para la solución de este problema se requiere que el usuario ingrese dos números; luego, que el sistema devuelva el resultado para saber si los números son amigos o no.

Entrada

- Números (n_1 , n_2)

Salida

- Resultado (r)

Diseño:**Interfaz de usuario**

The screenshot shows a web browser window titled 'Problema 63 - Windows Internet Explorer'. The address bar shows 'http://localhost/Cap06/Problema63.php'. The browser has a menu bar with 'Archivo', 'Edición', 'Ver', 'Favoritos', 'Herramientas', and 'Ayuda'. Below the menu bar is a search bar and a tab labeled 'Problema 63'. The main content area displays the following form:

Problema 63

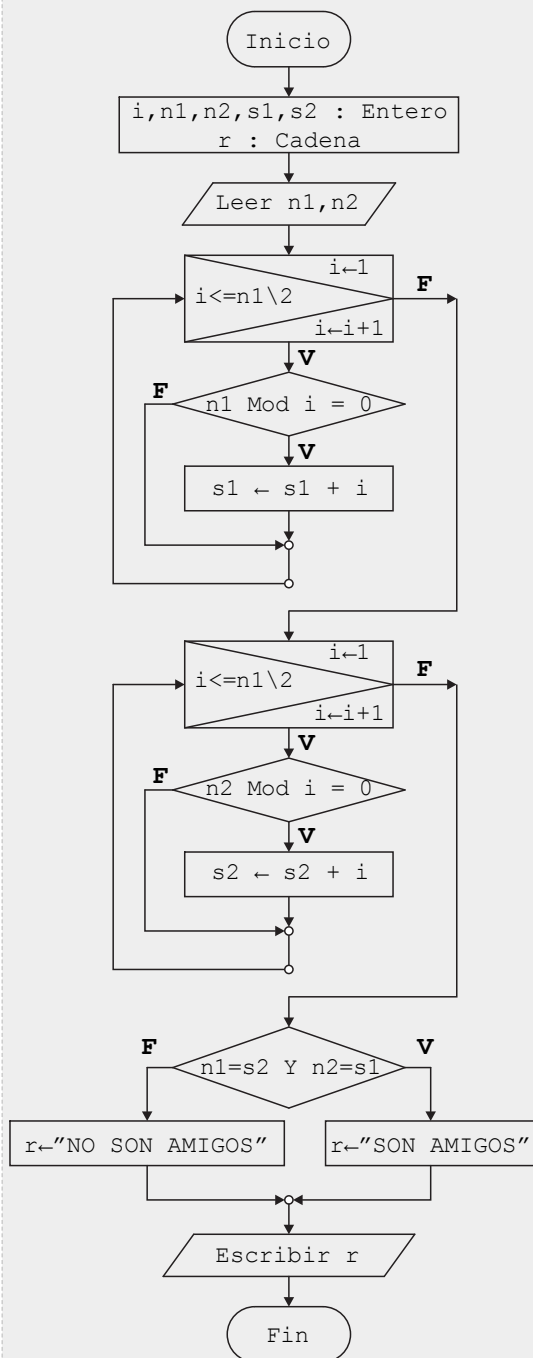
Numero 1

Numero 2

Resultado

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

i, n1, n2, s1, s2 : Entero

r : Cadena

//Entrada

Leer n1, n2

//Proceso

Para i ← 1 Hasta n\2 Inc 1

Si n1 Mod i = 0 Entonces

s1 ← s1 + i

Fin Si

Fin Para

Para i ← 1 Hasta n\2 Inc 1

Si n2 Mod i = 0 Entonces

s2 ← s2 + i

Fin Si

Fin Para

Si n1=s2 Y n2=s1 Entonces

r ← "SON AMIGOS"

SiNo

r ← "NO SON AMIGOS"

Fin Si

//Salida

Escribir r

Fin

Codificación:

```

<?php
//Variables
$i = 0; $n1 = 0; $n2 = 0; $s1 = 0; $s2 = 0;
$r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n1 = (int)$_POST["txtn1"];
    $n2 = (int)$_POST["txtn2"];

    //Proceso
    for($i = 1; $i <= $n1/2; $i++){
        if($n1 % $i == 0)
            $s1 += $i;
    }

    for($i = 1; $i <= $n2/2; $i++){
        if($n2 % $i == 0)
            $s2 += $i;
    }

    if($n1 == $s2 && $n2 == $s1)
        $r = "SON AMIGOS";
    else
        $r = "NO SON AMIGOS";
}
?>
<html>
<head>
<title>Problema 63</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema63.php">
    <table width="251" border="0">
        <tr>
            <td colspan="2"><strong>Problema 63</strong> </td>
        </tr>
        <tr>
            <td width="85">Numero 1 </td>
            <td width="179">
                <input name="txtn1" type="text" id="txtn1" value="<?=$n1?>"
            </td>
        </tr>
        <tr>
            <td>Numero 2 </td>

```

```

        <td><input name="txtn2" type="text" id="txtn2" value="<?=$n2?>"
/></td>
    </tr>

    <tr>
        <td>Resultado</td>
        <td><input name="txtr2" type="text" class="TextoFondo" id="txtr2"
value="<?=$r?>" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 64

Enunciado: Escriba un algoritmo que calcule, la suma de la siguiente serie, hasta el número entero positivo N ingresado.

$$\frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \frac{4}{N}, \text{ por ejemplo si N es 3 entonces } \frac{1}{2} + \frac{2}{3} + \frac{7}{6} = 1,1666667$$

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema devuelva el resultado de la suma de quebrados.

Entrada

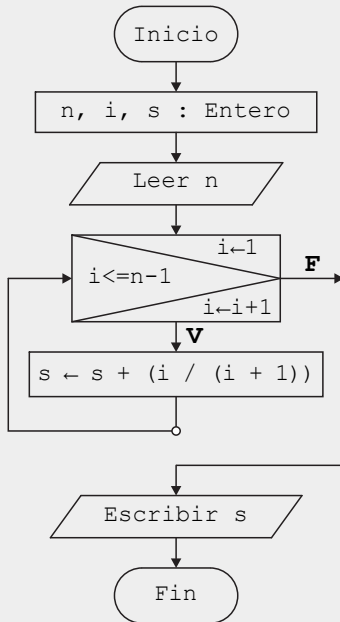
- Número (n)

Salida

- Suma (s)

Diseño:

Interfaz de usuario

Diagrama de flujo**Algoritmo****Inicio****//Variables**

i, n, s : Entero

//Entrada

Leer n

//Proceso

Para i ← 1 Hasta n-1 Inc 1

s ← s + (i / (i + 1))

Fin Para

//Salida

Escribir s

Fin**Codificación:**

```

<?php

//Variables
$n = 0; $i = 0; $s = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    for($i = 1; $i <= $n - 1; $i++)
        $s = $s + ($i / ($i + 1));
}
?>
<html>
<head>
<title>Problema 64</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema64.php">
    <table width="244" border="0">
        <tr>

```

```

        <td colspan="2"><strong>Problema 64</strong> </td>
    </tr>
    <tr>
        <td width="73">Numero </td>
        <td width="161">
            <input name="txtn" type="text" id="txtn" value="<?=$n?" />
        </td>
    </tr>
    <tr>
        <td>Suma</td>
        <td><input name="txtn2" type="text" class="TextoFondo" id="txtn2"
value="<?=$s?" /></td>
    </tr>

    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 65

Enunciado: Dado un rango numérico entero (número inicial y número final), obtener la cantidad de números positivos y negativos que existen en el rango.

Análisis: Para la solución de este problema se requiere que el usuario ingrese dos números; luego, que el sistema devuelva la cantidad de números positivos y negativos.

Entrada

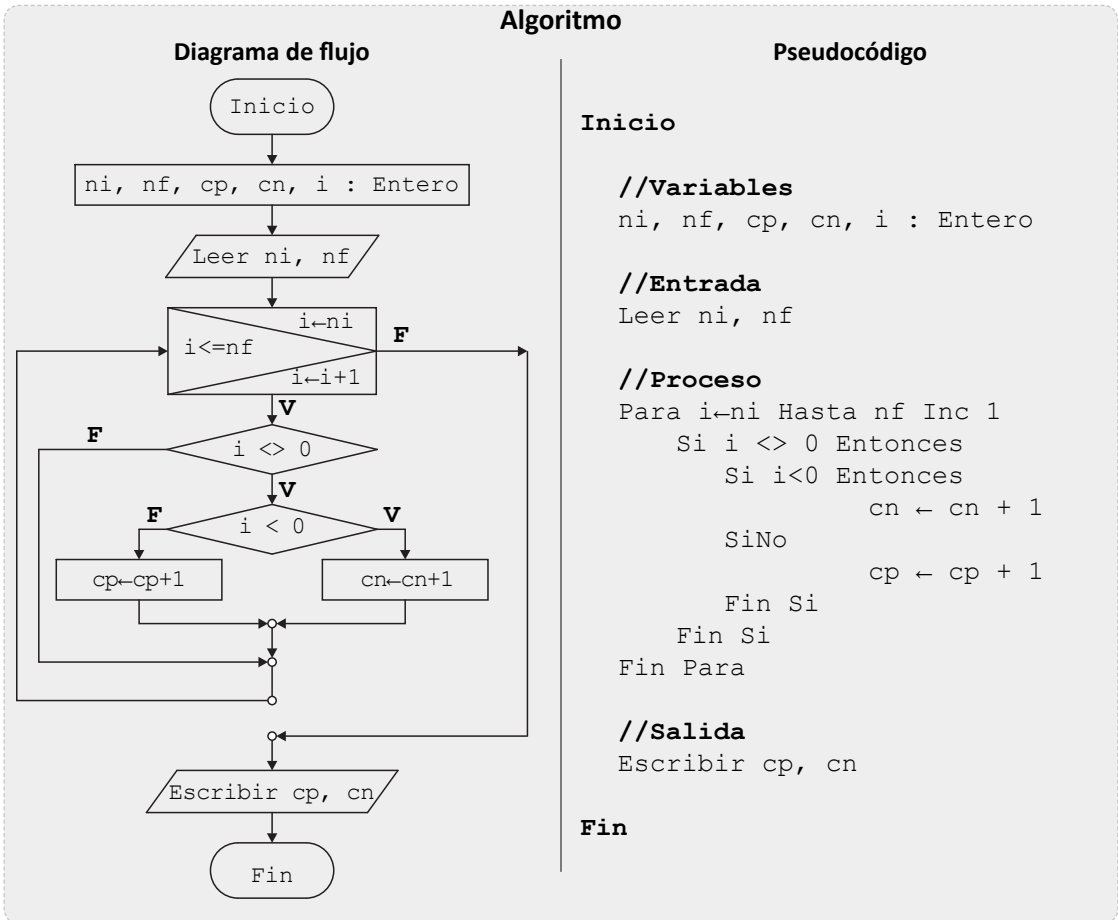
- Número inicial (ni)
- Número final (nf)

Salida

- Cantidad positivos (cp)
- Cantidad negativos (cn)

Diseño:

Interfaz de usuario

**Codificación:**

```

<?php
//Variables
$ni = 0; $nf = 0; $cp = 0; $cn = 0; $i = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $ni = (int)$_POST["txtni"];
    $nf = (int)$_POST["txtnf"];

    //Proceso
    for($i = $ni; $i <= $nf; $i++) {
        if($i != 0) {
            if($i < 0)
                $cn += 1;
            else
                $cp += 1;
        }
    }
}
?>
<html>
<head>
<title>Problema 65</title>

```

```

<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema65.php">
    <table width="264" border="0">
        <tr>
            <td colspan="2"><strong>Problema 65</strong> </td>
        </tr>
        <tr>
            <td width="105">Num. Inicial </td>
            <td width="149">
                <input name="txtni" type="text" id="txtni" value="<?=$ni?>"
            </td>
        </tr>
        <tr>
            <td>Num. Final </td>
            <td><input name="txtnf" type="text" id="txtnf" value="<?=$nf?>"
        </td>
        </tr>

        <tr>
            <td>Cant. Positivos </td>
            <td><input name="txtcp" type="text" class="TextoFondo" id="txtcp"
value="<?=$cp?>" /></td>
        </tr>
        <tr>
            <td>Cant. Negativos </td>
            <td><input name="txtcn" type="text" class="TextoFondo" id="txtcn"
value="<?=$cn?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;   </td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 66

Enunciado: Hallar cuántos múltiplos de M hay en un rango de números enteros.

Análisis: Para la solución de este problema se requiere que el usuario ingrese tres números (inicial, final y múltiplo); luego, que el sistema devuelva la cantidad de múltiplos que hay en el rango.

Entrada

- Número inicial (ni)
- Número final (nf)
- Número múltiplo

Salida

- Cantidad (c)

Diseño:**Interfaz de usuario**

Problema 66 - Windows Internet Explorer

http://localhost/Cap06/Problema66.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 66

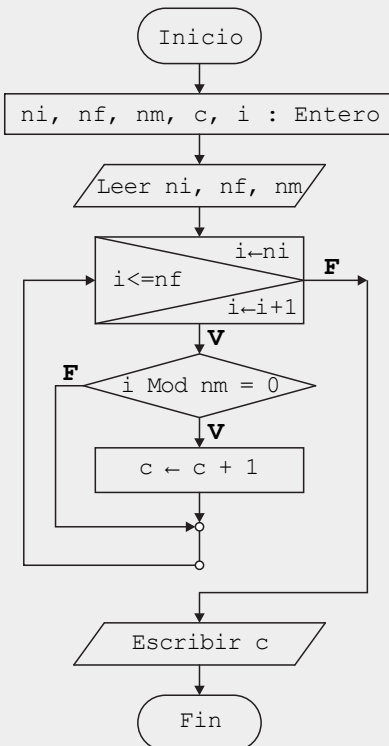
Problema 66

Num. Inicial

Num. Final

Num. Multiplo

Cantidad

Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio****//Variables**

ni, nf, nm, c, i : Entero

//Entrada

Leer ni, nf, nm

//Proceso

Para i ← ni Hasta nf Inc 1
 Si i Mod nm = 0 Entonces
 c ← c + 1

Fin Si

Fin Para

//Salida

Escribir c

Fin

Codificación:

```

<?php

//Variables
$ni = 0; $nf = 0; $nm = 0; $c = 0; $i = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $ni = (int)$_POST["txtni"];
    $nf = (int)$_POST["txtnf"];
    $nm = (int)$_POST["txtnm"];

    //Proceso
    for($i = $ni; $i<=$nf; $i++){
        if($i % $nm == 0)
            $c += 1;
    }
}
?>
<html>
<head>
<title>Problema 66</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema66.php">
    <table width="264" border="0">
        <tr>
            <td colspan="2"><strong>Problema 66</strong> </td>
        </tr>
        <tr>
            <td width="105">Num. Inicial </td>
            <td width="149">
                <input name="txtni" type="text" id="txtni" value="<?=$ni?>"
            />
            </td>
        </tr>
        <tr>
            <td>Num. Final </td>
            <td><input name="txtnf" type="text" id="txtnf" value="<?=$nf?>"
        /></td>
        </tr>

        <tr>
            <td>Num. Multiplo </td>

```

```

<td><input name="txtnm" type="text" id="txtnm" value="<?=$nm?>"
/></td>
</tr>
<tr>
<td>Cantidad</td>
<td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" /></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>
<input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
</td>
</tr>
</table>
</form>
</body>
</html>

```

Problema n.º 67

Enunciado: Crear un algoritmo para hallar el factorial de un número; el factorial es el producto de todos los números consecutivos desde la unidad hasta el número. Por ejemplo, factorial de 3! (se denota !) es $1 \times 2 \times 3 = 6$.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema devuelva el factorial del número.

Entrada

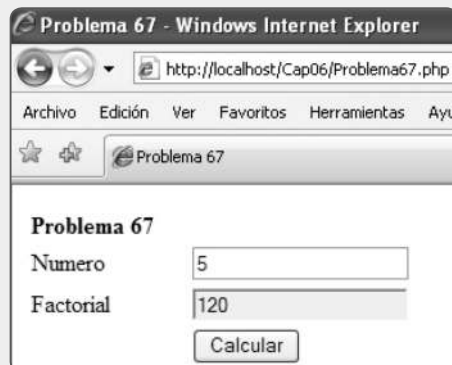
- Número (n)

Salida

- Factorial (f)

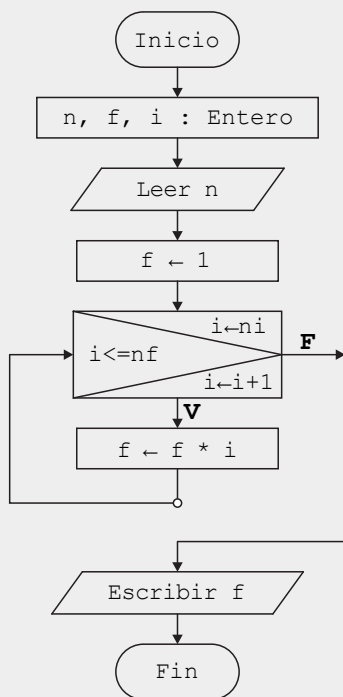
Diseño:

Interfaz de usuario



Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio**//Variables**

n, f, i : Entero

//Entrada

Leer n

//Proceso

f ← 1

Para i←1 Hasta n Inc 1

f ← f * i

Fin Para

//Salida

Escribir f

Fin

Codificación:

```

<?php

//Variables
$n = 0; $f = 0; $i = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $f = 1;
    for($i = 1; $i<=$n; $i++)
        $f *= $i;
}
?>
<html>
<head>
<title>Problema 67</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema67.php">

```

```

<table width="264" border="0">
  <tr>
    <td colspan="2"><strong>Problema 67</strong> </td>
  </tr>
  <tr>
    <td width="105">Numero</td>
    <td width="149">
      <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
    </td>
  </tr>
  <tr>
    <td>Factorial</td>
    <td><input name="txtf" type="text" class="TextoFondo" id="txtf"
value="<?=$f?>" /></td>
  </tr>

  <tr>
    <td>&nbsp;</td>
    <td>
      <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
    </td>
  </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 68

Enunciado: Determine si un número es primo; se dice que un número es primo si es divisible entre 1 y entre sí mismo.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema devuelva si el número es o no primo.

Entrada

- Número (n)

Salida

- Respuesta (r)

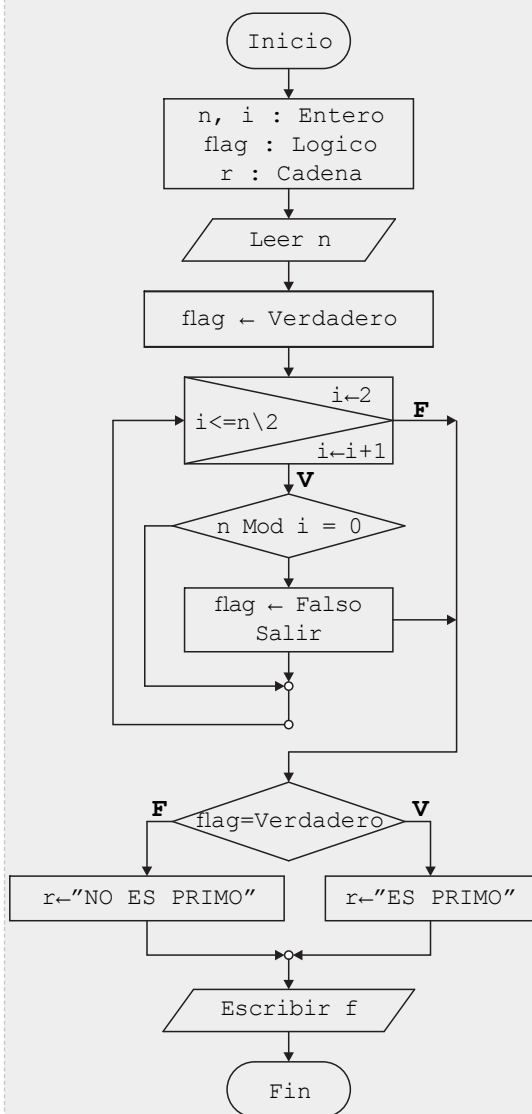
Diseño:

Interfaz de usuario

The screenshot shows a web browser window titled "Problema 68 - Windows Internet Explorer". The address bar displays the URL "http://localhost/Cap06/Problema68.php". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The page content is titled "Problema 68" and contains a form with two input fields: "Numero" with the value "7" and "Respuesta" with the value "ES PRIMO". Below these fields is a "Calcular" button.

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

```

n, i : Entero
flag : Logico
r : Cadena

```

//Entrada

```

Leer n

```

//Proceso

```

flag ← Verdadero
Para i←1 Hasta n\2 Inc 1
    Si n Mod i = 0 Entonces
        flag ← Falso
        Salir
    Fin Si
Fin Para

```

```

Si flag = Verdadero Entonces
    r ← "ES PRIMO"
SiNo
    r ← "NO ES PRIMO"
Fin Si

```

//Salida

```

Escribir r

```

Fin

Codificación:

```

<?php

//Variables
$n = 0; $i = 0;
$flag = false;
$r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    $flag = true;
    $i = 2;
    for($i = 2; $i<=$n/2; $i++){

```

```

        if($n % $i == 0) {
            $flag = false;
            break;
        }
    }

    if($flag)
        $r = "ES PRIMO";
    else
        $r = "NO ES PRIMO";
}
?>
<html>
<head>
<title>Problema 68</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema68.php">
    <table width="264" border="0">
        <tr>
            <td colspan="2"><strong>Problema 68</strong> </td>
        </tr>
        <tr>
            <td width="105">Numero</td>
            <td width="149">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Respuesta</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 69

Enunciado: Determine cuántos números primos hay en los primeros N números enteros positivos.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema devuelva la cantidad de números primos; por ejemplo, si ingresa 7, hay 4 números primos 1, 3, 5 y 7.

Entrada

- Número (n)

Salida

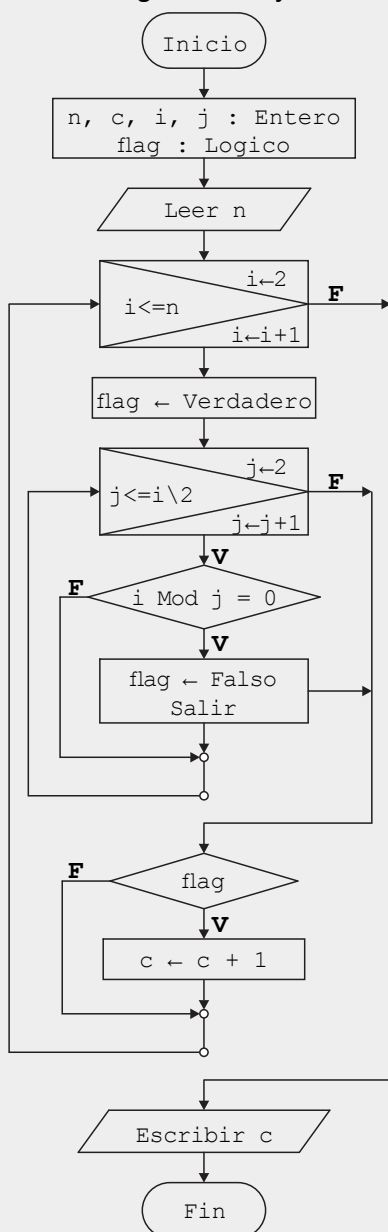
- Cantidad (c)

Diseño: Interfaz de usuario



Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

n, c, i, j : Entero
flag : Logico

//Entrada

Leer n

//Proceso

```

Para i ← 2 Hasta n Inc 1
    flag ← Verdadero
    Para j ← 2 Hasta i \ 2 Inc 1
        Si i Mod j = 0 Entonces
            flag ← Falso
            Salir
    Fin Si
Fin Para

```

```

Si flag Entonces
    c ← c + 1
Fin Si
Fin Para

```

```

Si flag Entonces
    c ← c + 1
Fin Si
Fin Para

```

```

Fin Para

```

//Salida

Escribir c

Fin

Codificación:

```

<?php

//Variables
$n = 0; $c = 0; $i = 0; $j = 0;
$flag = false;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];

    //Proceso
    for($i = 2; $i<=$n; $i++){
        $flag = true;
        for($j = 2; $j <= $i / 2; $j ++){
            if($i % $j == 0) {
                $flag = false;
                break;
            }
        }
        if($flag) {
            $c += 1;
            $flag = true;
        }
    }
}
?>
<html>
<head>
<title>Problema 69</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema69.php">
    <table width="264" border="0">
        <tr>
            <td colspan="2"><strong>Problema 69</strong> </td>
        </tr>
        <tr>
            <td width="105">Numero</td>
            <td width="149">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
    </table>

```

```
<tr>
  <td>Cantidad</td>
  <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" /></td>
</tr>

<tr>
  <td>&nbsp;</td>
  <td>
    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
  </td>
</tr>
</table>
</form>
</body>
</html>
```

Problema n.º 70

Enunciado: Dado un número y un divisor, determine cuál es el número múltiplo antecesor al número ingresado, por ejemplo, si ingresa $N = 21$ y $D = 3$, entonces $R = 18$ porque es el número múltiplo de 3 antecesor de 21.

Análisis: Para la solución de este problema se requiere que el usuario ingrese un número; luego, que el sistema devuelva el número múltiplo antecesor.

Entrada

- Número (n)
- Divisor (d)

Salida

- Respuesta (r)

Diseño:

Interfaz de usuario

Problema 70 - Windows Internet Explorer

http://localhost/Cap06/Problema70.php

Archivo Edición Ver Favoritos Herramientas Ay

☆ Problema 70

Problema 70

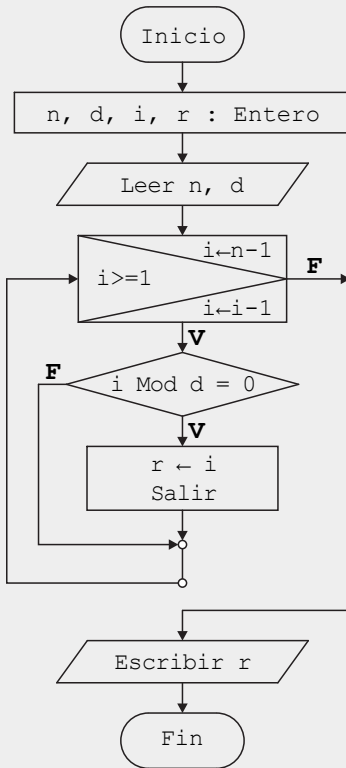
Numero

Divisor

Respuesta

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

n, d, i, r : Entero

//Entrada

Leer n, d

//Proceso

Para i ← n-1 Hasta 1 Inc -1

 Si i Mod d = 0 Entonces

 r ← i

 Salir

 Fin Si

Fin Para

//Salida

Escribir r

Fin

Codificación:

```

<?php

//Variables
$n = 0; $d = 0; $i = 0; $r = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = (int)$_POST["txtn"];
    $d = (int)$_POST["txtd"];

    //Proceso
    for($i = $n - 1; $i >= 1; $i--=1) {
        if($i % $d == 0){
            $r = $i;
            break;
        }
    }
}
?>
<html>
<head>

```

```

<title>Problema 70</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema70.php">
    <table width="264" border="0">
        <tr>
            <td colspan="2"><strong>Problema 70</strong> </td>
        </tr>
        <tr>
            <td width="105">Numero</td>
            <td width="149">
                <input name="txtn" type="text" id="txtn" value="<?=$n?" />
            </td>
        </tr>
        <tr>
            <td>Divisor</td>
            <td><input name="txtd" type="text" id="txtd" value="<?=$d?" /></td>
        </tr>
        <tr>
            <td>Respuesta</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr" value="<?=$r?" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular" value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

6.4 Problemas propuestos

Los siguientes problemas le servirán para medir su aprendizaje, es importante que los resuelva.

Propuesto n.º 41

Enunciado: Calcule la suma de los cuadrados y cubos de los N primeros números naturales.

Propuesto n.º 42

Enunciado: Obtener la suma y la cantidad de los números divisibles por 3 y 5 a la vez, de los N primeros números naturales.

Propuesto n.º 43

Enunciado: Dado un rango numérico entero positivo a y b, obtener la suma y la cantidad de los números pares, impares y múltiplos de 3.

Propuesto n.º 44

Enunciado: Calcule la suma y la cantidad de números de la serie de fibonacci, menores a N. La serie de fibonacci es una secuencia de números cuya característica es que cada número de la serie debe ser igual a la suma de los 2 números anteriores; la serie empieza con 0 y 1, entonces, si el número N ingresado es 30, entonces la serie sería menor a 30. Esto equivale a 0 1 1 2 3 5 8 13 21, y lo que se pide es la suma y la cantidad de números de la serie.

Propuesto n.º 45

Enunciado: Dado un rango de números determine cuántos números capicúa hay.

Propuesto n.º 46

Enunciado: Dado la cantidad de cifras y un divisor, determine cuántos números múltiplos existen del divisor con dichas cifras.

Propuesto n.º 47

Enunciado: Calcule la suma de la siguiente serie.

$$s = \frac{1}{0!} + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots + \frac{x^n}{n!}$$

Propuesto n.º 48

Enunciado: Calcule de cuántas formas se pueden ordenar n objetos.

Propuesto n.º 49

Enunciado: Obtenga la cantidad de los números primos de n cifras.

Propuesto n.º 50

Enunciado: Obtenga la cantidad de los números capicúas de n cifras.

Capítulo 7

Estructuras de datos. Arreglos (vectores y matrices)

7.1 Introducción

En muchas situaciones se necesita procesar una colección de datos que están relacionados entre sí, por ejemplo, la lista de notas de los alumnos, los participantes de una carrera deportiva, etc. Procesar ese conjunto de datos en forma independiente con variables simples (primitivas), es tremendamente difícil, por eso los lenguajes de programación incorporan un mecanismo que facilita la manipulación y organización para una colección de datos llamada «estructura de datos».

Para explicar todo lo relacionado a estructura de datos se necesita escribir todo un libro que detalle los temas involucrados, para este capítulo solo se está considerando una parte básica e importante en la estructura de datos, llamada **array** (arreglos).

Vector					Matriz				
0	1	2	3	4		0	1	2	3
15	12	18	14	12	0	25	10	15	32
					1	52	10	4	18
					2	18	22	3	9

Las estructuras de datos están subdivididas por estáticas (espacio fijo establecido en memoria) y dinámicas (sin restricciones y limitaciones en el espacio usado en memoria).

Estructuras de datos estáticas

- Arrays (vectores y matrices)
- Cadenas
- Registros
- Ficheros

Estructuras de datos dinámicas

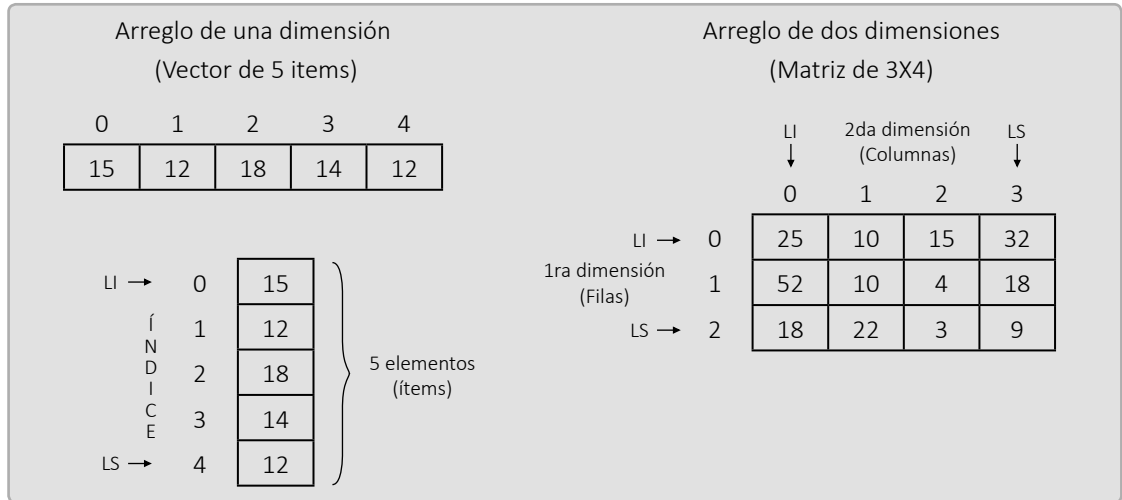
- Listas (pilas y colas)
- Listas enlazadas
- Árboles
- Grafos

La diferencia entre cada estructura es la forma de cómo se almacena y manipula el conjunto de datos, permitiendo así su eficiencia en el resultado de una operación sobre dichos datos.

7.2 Arrays (Arreglos)

Es un conjunto finito (tamaño fijo) y ordenado (usa un índice) de datos homogéneos (datos del mismo tipo). Los arreglos pueden ser de una dimensión (vector), dos dimensiones (matriz) y n dimensiones (multidimensional).

En todos los lenguajes de programación, los **arrays** usan un índice numérico para cada elemento que contiene, los cuales por lo general inician con el índice 0, llamado «Limite Inferior» (LI); y el último elemento tendrá el índice llamado «Limite Superior» (LS), que en sí es la cantidad de elementos del **array** menos 1.



7.3 Operaciones con **arrays**

Las operaciones son el procesamiento y el tratamiento individual de los elementos del **array**, las cuales son las siguientes:

- Asignación
- Lectura / escritura
- Recorrido
- Actualización (insertar, borrar, modificar)
- Ordenación
- Búsqueda

7.4 Creación de *arrays*

Para la creación de un *array* se requiere conocer el nombre, las dimensiones, el tamaño de elementos y el tipo de dato.

Pseudocódigo

```
//Array de una dimensión (Vector)
// 5 elementos LI = 0 y LS = 4
N[5] : Entero

//Array de dos dimensiones (Matriz)
// 3X4 elementos
// 1era Dim. LI = 0 y LS = 2
// 2da Dim. LI = 0 y LS = 3
N[3][4] : Entero
```

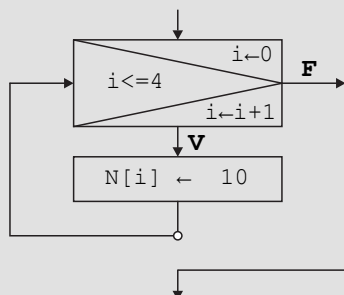
PHP

```
//Array de una dimensión (Vector)
$N = array();
$N[0] = 10;
$N[1] = 20;
...

//Array de dos dimensiones (Matriz)
$N = array();
$N[0][0] = 10;
$N[0][1] = 20;
...
```

7.5 Recorrido por los elementos del *array*

Para realizar un recorrido por cada elemento del *array* utilizaremos la estructura repetitiva «Para» (For). En el siguiente diagrama se tiene el **vector** N de 5 elementos, y se asigna el valor 10 a cada elemento.



Para i ← 0 **Hasta** 4 **Inc** +1

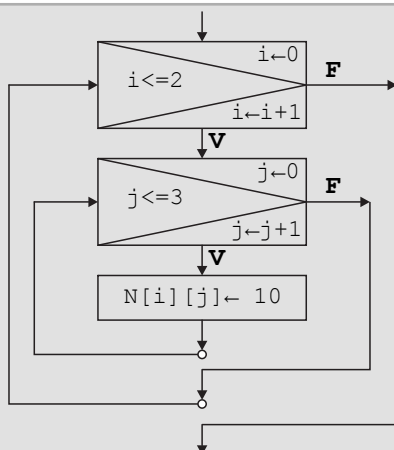
N[i] ← 10

Fin Para

Sintaxis PHP

```
for ($i=0;$i<=4;$i++) {
    $N[$i] = 10;
}
```

En el siguiente diagrama se tiene la **matriz** N de 3 x 4 elementos y se asigna el valor 10 a cada elemento.



Para i ← 0 **Hasta** 2 **Inc** +1

Para j ← 0 **Hasta** 3 **Inc** +1

N[i][j] ← 10

Fin Para

Fin Para

Sintaxis PHP

```
for ($i=0;$i<=2;$i++) {
    for ($j=0;$j<=3;$j++) {

        $N[$i][$j] = 10;

    }
}
```

Problema n.º 71

Enunciado: Dados 5 números, obtener la suma.

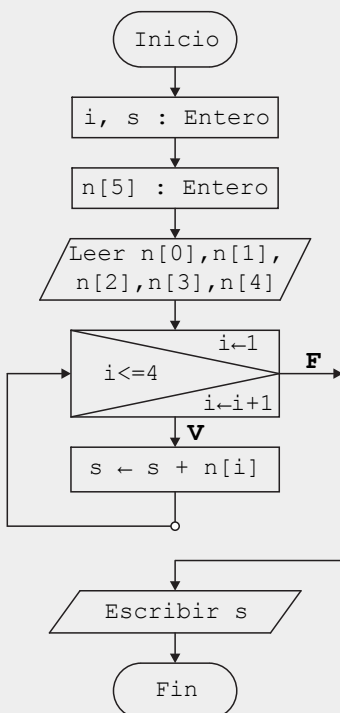
Análisis: Para la solución de este problema se requiere que el usuario ingrese 5 números; luego, que el sistema realice el proceso para devolver la suma.

Entrada

- 5 números $n[5]$

Salida

- Suma (s)

Diseño:**Interfaz de usuario**
Diagrama de flujo**Algoritmo****Inicio****//Variables**

i, s : Entero

//Arreglos (Vector)

$n[5]$: Entero

//Entrada

Leer $n[0], n[1], n[2], n[3], n[4]$

//Proceso

Para $i \leftarrow 0$ Hasta 4 Inc 1

$s \leftarrow s + n[i]$

Fin Para

//Salida

Escribir s

Fin

Codificación:

```

<?php

//Variables
$s = 0; $i = 0;

//Arreglos
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0] = (int)$_POST["txtn1"];
    $n[1] = (int)$_POST["txtn2"];
    $n[2] = (int)$_POST["txtn3"];
    $n[3] = (int)$_POST["txtn4"];
    $n[4] = (int)$_POST["txtn5"];

    //Proceso
    for($i = 0; $i <= 4; $i++)
        $s += $n[$i];
}
?>
<html>
<head>
<title>Problema 71</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema71.php">
    <table width="245" border="0">
        <tr>
            <td colspan="2"><strong>Problema 71</strong> </td>
        </tr>
        <tr>
            <td width="82">Numero 1 </td>
            <td width="153">
                <input name="txtn1" type="text" id="txtn1" value="<?=$n[0]?>"
size="10" /> </td>
            </tr>
            <tr>
                <td>Numero 2 </td>
                <td><input name="txtn2" type="text" id="txtn2" value="<?=$n[1]?>"
size="10" /></td>
            </tr>

            <tr>
                <td>Numero 3 </td>
                <td><input name="txtn3" type="text" id="txtn3" value="<?=$n[2]?>"
size="10" /></td>
            </tr>
            <tr>

```

```

        <td>Numero 4 </td>
        <td><input name="txtn4" type="text" id="txtn4" value="<?=$n[3]?>"
size="10" /></td>
    </tr>
    <tr>
        <td>Numero 5 </td>
        <td><input name="txtn5" type="text" id="txtn5" value="<?=$n[4]?>"
size="10" /></td>
    </tr>
    <tr>
        <td>Suma</td>
        <td><input name="txts" type="text" class="TextoFondo" id="txts"
value="<?=$s?>" size="10" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 72

Enunciado: Dados 5 números, obtener el número mayor.

Análisis: Para la solución de este problema se requiere que el usuario ingrese 5 números; luego, que el sistema realice el proceso para devolver el mayor.

Entrada

- 5 números $n[5]$

Salida

- Mayor (m)

Diseño:

Interfaz de usuario

Problema 72 - Windows Internet Explorer

http://localhost/Cap07/Problema72.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 72

Problema 72

Numero 1 5

Numero 2 2

Numero 3 34

Numero 4 55

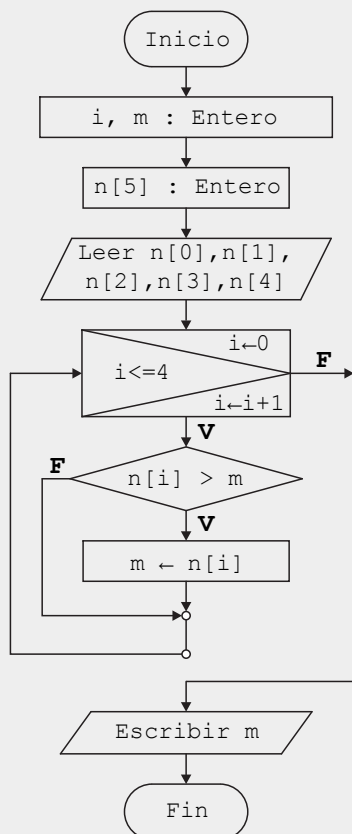
Numero 5 3

Mayor 55

Calcular

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

i, m : Entero

//Arreglos (Vector)

n[5] : Entero

//Entrada

Leer n[0], n[1], n[2], n[3], n[4]

//Proceso

Para i ← 0 Hasta 4 Inc 1

Si n[i] > m Entonces

m ← n[i]

Fin Si

Fin Para

//Salida

Escribir m

Fin

Codificación:

```

<?php

//Variables
$i = 0; $m = 0;

//Arreglos
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0] = (int)$_POST["txtn1"];
    $n[1] = (int)$_POST["txtn2"];
    $n[2] = (int)$_POST["txtn3"];
    $n[3] = (int)$_POST["txtn4"];
    $n[4] = (int)$_POST["txtn5"];

    //Proceso
    for($i = 0; $i <= 4; $i++){
        if($n[$i] > $m)
            $m = $n[$i];
    }
}
?>

```

```

<html>
<head>
<title>Problema 72</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema72.php">
    <table width="165" border="0">
        <tr>
            <td colspan="2"><strong>Problema 72</strong> </td>
        </tr>
        <tr>
            <td width="83">Numero 1 </td>
            <td width="72">
                <input name="txtn1" type="text" id="txtn1" value="<?=$n[0]?>"
size="10" /> </td>
            </tr>
            <tr>
                <td>Numero 2 </td>
                <td><input name="txtn2" type="text" id="txtn2" value="<?=$n[1]?>"
size="10" /></td>
            </tr>

            <tr>
                <td>Numero 3 </td>
                <td><input name="txtn3" type="text" id="txtn3" value="<?=$n[2]?>"
size="10" /></td>
            </tr>
            <tr>
                <td>Numero 4 </td>
                <td><input name="txtn4" type="text" id="txtn4" value="<?=$n[3]?>"
size="10" /></td>
            </tr>
            <tr>
                <td>Numero 5 </td>
                <td><input name="txtn5" type="text" id="txtn5" value="<?=$n[4]?>"
size="10" /></td>
            </tr>
            <tr>
                <td>Mayor</td>
                <td><input name="txtm" type="text" class="TextoFondo" id="txtm"
value="<?=$m?>" size="10" /></td>
            </tr>
            <tr>
                <td>&nbsp;</td>
                <td>
                    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" /> </td>
            </tr>
        </table>
    </form>
</body>
</html>

```

Problema n.º 73

Enunciado: Dados 5 números y un divisor, determinar cuántos números múltiplos hay del divisor en los 5 números ingresados.

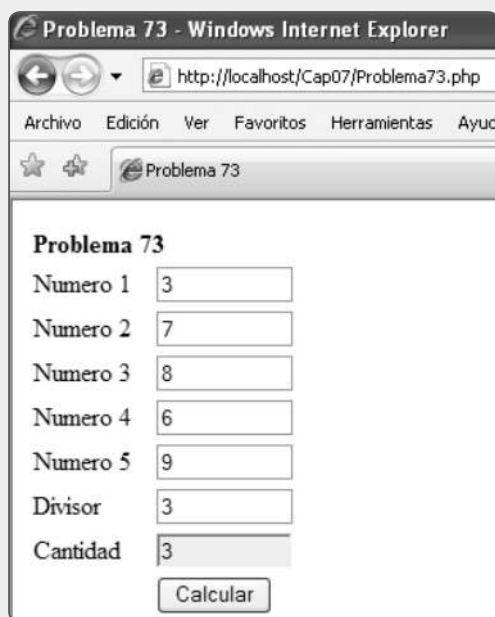
Análisis: Para la solución de este problema se requiere que el usuario ingrese 5 números; luego, que el sistema procese y devuelva la cantidad de números múltiplos que hay.

Entrada

- 5 números (n[5])
- Divisor (d)

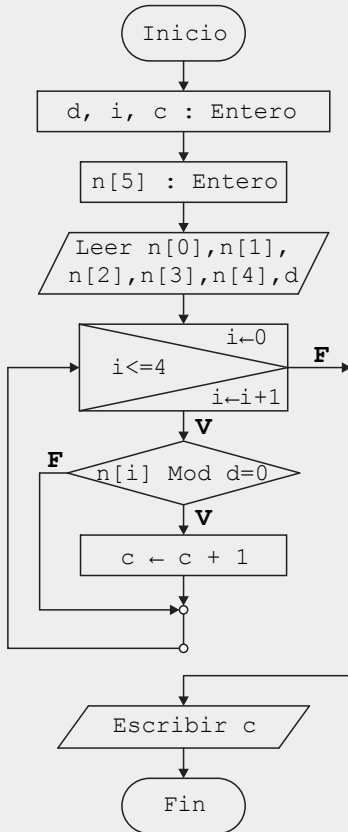
Salida

- Cantidad (c)

Diseño:**Interfaz de usuario**

The screenshot shows a web browser window titled "Problema 73 - Windows Internet Explorer". The address bar displays "http://localhost/Cap07/Problema73.php". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayud". The page content is titled "Problema 73" and contains the following form elements:

- Five input fields labeled "Numero 1" through "Numero 5" with values 3, 7, 8, 6, and 9 respectively.
- An input field labeled "Divisor" with the value 3.
- An input field labeled "Cantidad" with the value 3.
- A "Calcular" button.

Algoritmo**Diagrama de flujo****Pseudocódigo**

Inicio

//Variables**d, i, c : Entero****//Arreglos (Vector)****n[5] : Entero****//Entrada**

Leer n[0],n[1],n[2],n[3],n[4],d

//Proceso

Para i←0 Hasta 4 Inc 1

Si n[i] Mod d = 0 Entonces

c ← c + 1

Fin Si

Fin Para

//Salida

Escribir c

Fin**Codificación:**

```

<?php

//Variables
$d = 0; $i = 0; $c = 0;

//Arreglos
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0] = (int)$_POST["txtn1"];
    $n[1] = (int)$_POST["txtn2"];
    $n[2] = (int)$_POST["txtn3"];
    $n[3] = (int)$_POST["txtn4"];
    $n[4] = (int)$_POST["txtn5"];
    $d = (int)$_POST["txtd"];

    //Proceso
    for($i = 0; $i <= count($n) - 1; $i++){
        if($n[$i] % $d == 0)
            $c += 1;
    }
}

```

```

    }
}
?>
<html>
<head>
<title>Problema 73</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema73.php">
    <table width="177" border="0">
        <tr>
            <td colspan="2"><strong>Problema 73</strong> </td>
        </tr>
        <tr>
            <td width="74">Numero 1 </td>
            <td width="93">
                <input name="txtn1" type="text" id="txtn1" value="<?=$n[0]?>"
size="10" /> </td>
            </tr>
            <tr>
                <td>Numero 2 </td>
                <td><input name="txtn2" type="text" id="txtn2" value="<?=$n[1]?>"
size="10" /></td>
            </tr>

            <tr>
                <td>Numero 3 </td>
                <td><input name="txtn3" type="text" id="txtn3" value="<?=$n[2]?>"
size="10" /></td>
            </tr>
            <tr>
                <td>Numero 4 </td>
                <td><input name="txtn4" type="text" id="txtn4" value="<?=$n[3]?>"
size="10" /></td>
            </tr>
            <tr>
                <td>Numero 5 </td>
                <td><input name="txtn5" type="text" id="txtn5" value="<?=$n[4]?>"
size="10" /></td>
            </tr>
            <tr>
                <td>Divisor</td>
                <td><input name="txtd" type="text" id="txtd" value="<?=$d?>"
size="10" /></td>
            </tr>
            <tr>
                <td>Cantidad</td>

```

```

        <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" size="10" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 74

Enunciado: Dados 5 números, obtener la cantidad de números primos ingresados.

Análisis: Para la solución de este problema se requiere que el usuario ingrese 5 números; luego, que el sistema procese y devuelva la cantidad de números primos.

Entrada

- 5 números (n[5])

Salida

- Cantidad (c)

Diseño:

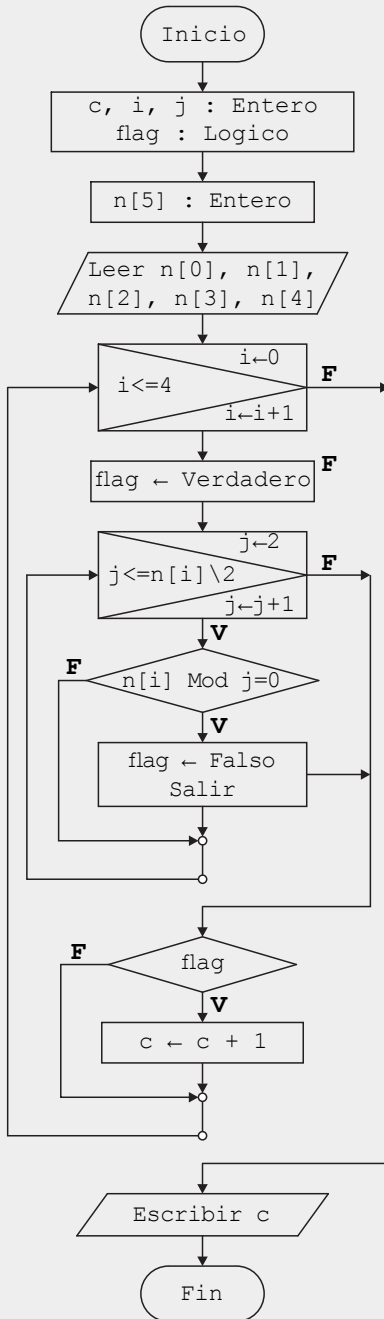
Interfaz de usuario

The screenshot shows a web browser window titled "Problema 74 - Windows Internet Explorer". The address bar shows "http://localhost/Cap07/Problema74.php". The browser has a menu bar with "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayu...". The page content is titled "Problema 74" and contains the following form elements:

- Numero 1: 7
- Numero 2: 3
- Numero 3: 6
- Numero 4: 4
- Numero 5: 11
- Cantidad: 3
- Calcular button

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio**//Variables**

c, i, j : Entero

flan : Logico

//Arreglos (Vector)

n[5] : Entero

//Entrada

Leer n[0], n[1], n[2], n[3], n[4]

//Proceso

Para i ← 0 Hasta 4 Inc 1

flag ← Verdadero

Para j ← 2 Hasta n[i] \ 2 Inc 1

Si n[i] Mod j = 0 Entonces

flag ← Falso

Salir

Fin Si

Fin Para

Si flag Entonces

c ← c + 1

Fin Si

Fin Para

//Salida

Escribir c

Fin

Codificación:

```

<?php

//Variables
$c = 0; $i = 0; $j = 0;
$flag = false;

//Arreglos
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0] = (int)$_POST["txtn1"];
    $n[1] = (int)$_POST["txtn2"];
    $n[2] = (int)$_POST["txtn3"];
    $n[3] = (int)$_POST["txtn4"];
    $n[4] = (int)$_POST["txtn5"];

    //Proceso
    for($i = 0; $i <= 4; $i++){
        $flag = true;
        for($j = 2; $j <= $n[$i]/2; $j++){
            if($n[$i] % $j == 0){
                $flag = false;
                break;
            }
        }
        if($flag)
            $c += 1;
    }
}
?>
<html>
<head>
<title>Problema 74</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema74.php">
    <table width="147" border="0">
        <tr>
            <td colspan="2"><strong>Problema 74</strong> </td>
        </tr>
        <tr>
            <td width="75">Numero 1 </td>
            <td width="62">
                <input name="txtn1" type="text" id="txtn1" value="<?=$n[0]?>"
                size="5" />
            </td>
        </tr>
    </table>
</form>

```

```

        </tr>
        <tr>
            <td>Numero 2 </td>
            <td><input name="txtn2" type="text" id="txtn2" value="<?=$n[1]?>"
size="5" /></td>
        </tr>

        <tr>
            <td>Numero 3 </td>
            <td><input name="txtn3" type="text" id="txtn3" value="<?=$n[2]?>"
size="5" /></td>
        </tr>
        <tr>
            <td>Numero 4 </td>
            <td><input name="txtn4" type="text" id="txtn4" value="<?=$n[3]?>"
size="5" /></td>
        </tr>
        <tr>
            <td>Numero 5 </td>
            <td><input name="txtn5" type="text" id="txtn5" value="<?=$n[4]?>"
size="5" /></td>
        </tr>

        <tr>
            <td>Cantidad</td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" size="5" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 75

Enunciado: Busque un número en 5 números ingresados, determine la posición y si existe o no el número buscado, use el método de búsqueda secuencial.

Análisis: Para la solución de este problema se requiere que el usuario ingrese 7 números; luego, que el sistema devuelva la respuesta, si existe o no el número y la posición del número encontrado.

Entrada

- 5 números (n[5])
- Numero a buscar (nb)

Salida

- Respuesta (r)
- Posición (p)

Diseño: Interfaz de usuario

Problema 75 - Windows Internet Explorer

http://localhost/Cap07/Problema75.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 75

Problema 75

Numero 1

Numero 2

Numero 3

Numero 4

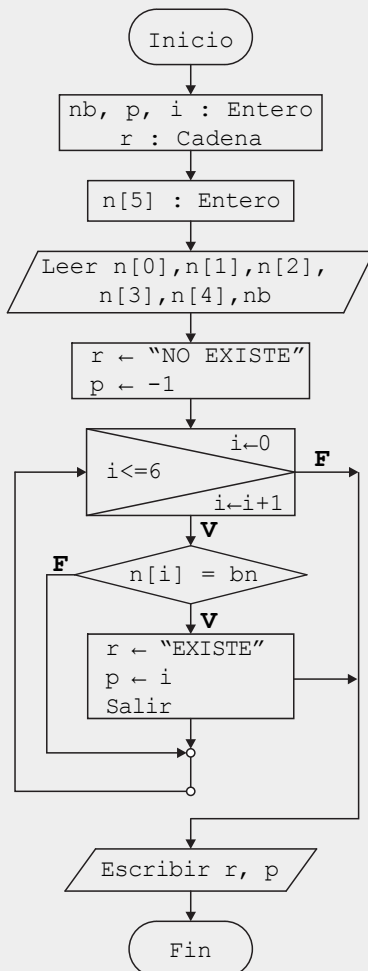
Numero 5

Num. a buscar

Respuesta

Posicion

Diagrama de flujo



Algoritmo

Pseudocódigo

Inicio

//Variables

nb, p, i : Entero
r : Cadena

//Arreglos (Vector)

n[5] : Entero

//Entrada

Leer n[0],n[1],n[2],n[3],n[4],
nb

//Proceso

```

r ← "NO EXISTE"
p ← -1
Para i ← 0 Hasta 6 Inc 1
    Si n[i] = nb Entonces
        r ← "EXISTE"
        p ← i
    Salir
Fin Si
Fin Para
  
```

//Salida

Escribir r, p

Fin

Codificación:

```

<?php

//Variables
$nb = 0; $p = 0; $i = 0;
$r = "";

//Arreglos
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0] = (int)$_POST["txtn1"];
    $n[1] = (int)$_POST["txtn2"];
    $n[2] = (int)$_POST["txtn3"];
    $n[3] = (int)$_POST["txtn4"];
    $n[4] = (int)$_POST["txtn5"];
    $nb = (int)$_POST["txtnb"];

    //Proceso
    $r = "NO EXISTE";
    $p = -1;
    for($i = 0 ; $i <= count($n) -1; $i++) {
        if($n[$i] == $nb){
            $r = "EXISTE";
            $p = $i;
            break;
        }
    }
}
?>
<html>
<head>
<title>Problema 75</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema75.php">
    <table width="184" border="0">
        <tr>
            <td colspan="2"><strong>Problema 75</strong> </td>
        </tr>
        <tr>
            <td width="99">Numero 1 </td>
            <td width="75">
                <input name="txtn1" type="text" id="txtn1" value="<?=$n[0]?>"
                size="5" />
            </td>
        </tr>
        <tr>
            <td>Numero 2 </td>
            <td><input name="txtn2" type="text" id="txtn2" value="<?=$n[1]?>"
                size="5" /></td>
        </tr>
    </table>

```

```

        <tr>
            <td>Numero 3 </td>
            <td><input name="txtn3" type="text" id="txtn3" value="<?=$n[2]?>"
size="5" /></td>
        </tr>
        <tr>
            <td>Numero 4 </td>
            <td><input name="txtn4" type="text" id="txtn4" value="<?=$n[3]?>"
size="5" /></td>
        </tr>
        <tr>
            <td>Numero 5 </td>
            <td><input name="txtn5" type="text" id="txtn5" value="<?=$n[4]?>"
size="5" /></td>
        </tr>

        <tr>
            <td>Num. a buscar </td>
            <td><input name="txtnb" type="text" id="txtnb" value="<?=$nb?>"
size="5" /></td>
        </tr>
        <tr>
            <td>Respuesta</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" size="10" /></td>
        </tr>
        <tr>
            <td>Posicion</td>
            <td><input name="txtp" type="text" class="TextoFondo" id="txtp"
value="<?=$p?>" size="5" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 76

Enunciado: Lea 4 números y almacénelos en un vector de llamado A; lo mismo con otros 4 números en un vector llamado B, y determine cuantos números de A se encuentran en B.

Análisis: Para la solución de este problema se requiere que el usuario ingrese 8 números; luego, que el sistema devuelve la cantidad.

Entrada

- 4 números (a[4])
- 4 números (b[4])

Salida

- Cantidad (c)

Diseño: Interfaz de usuario

Problema 76 - Windows Internet Explorer

http://localhost/Cap07/Problema76.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 76

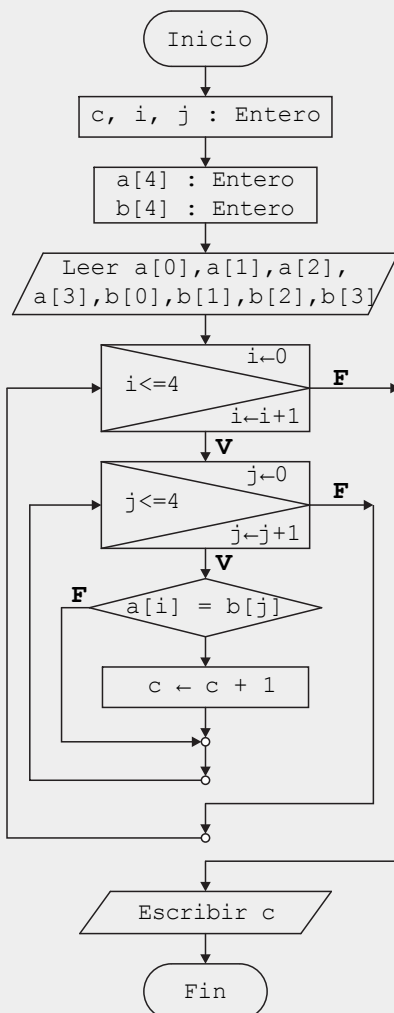
Problema 76

Arreglo A		Arreglo B	
Numero 1	3	Numero 1	5
Numero 2	5	Numero 2	8
Numero 3	2	Numero 3	6
Numero 4	1	Numero 4	3

Cantidad

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

c, i, j : Entero

//Arreglos (Vector)

a[4], b[4] : Entero

//Entrada

Leer a[0],a[1],a[2],a[3],
b[0],b[1],b[2],b[3]

//Proceso

```

Para i←0 Hasta 4 Inc 1
    Para j←0 Hasta 4 Inc 1
        Si a[i]=b[j] Entonces
            c ← c + 1
        Fin Si
    Fin Para
Fin Para
  
```

//Salida

Escribir c

Fin

Codificación:

```

<?php
//Variables
$c = 0; $i = 0; $j = 0;

//Arreglos
$a = array();
$b = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $a[0] = (int)$_POST["txta1"];
    $a[1] = (int)$_POST["txta2"];
    $a[2] = (int)$_POST["txta3"];
    $a[3] = (int)$_POST["txta4"];
    $b[0] = (int)$_POST["txtb1"];
    $b[1] = (int)$_POST["txtb2"];
    $b[2] = (int)$_POST["txtb3"];
    $b[3] = (int)$_POST["txtb4"];

    //Proceso
    for($i = 0; $i <= count($a) - 1; $i++) {
        for($j = 0; $j <= count($b) - 1; $j++) {
            if($a[$i] == $b[$j])
                $c += 1;
        }
    }
}
?>
<html>
<head>
<title>Problema 76</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema76.php">
    <table width="287" border="0">
        <tr>
            <td colspan="4"><strong>Problema 76</strong> </td>
        </tr>
        <tr>
            <td><strong>Arreglo A</strong> </td>
            <td>&nbsp;</td>
            <td width="74"><strong>Arreglo B </strong></td>
            <td width="43">&nbsp;</td>
        </tr>
        <tr>
            <td width="84">Numero 1 </td>
            <td width="68">
                <input name="txta1" type="text" id="txta1" value="<?=$a[0]?>"
                size="5" /> </td>
            <td>Numero 1 </td>
            <td><input name="txtb1" type="text" id="txtb1" value="<?=$b[0]?>"
                size="5" /> </td>
        </tr>
        <tr>
            <td>Numero 2 </td>

```

```

        <td><input name="txta2" type="text" id="txta2" value="<?=$a[1]?>"
size="5" /></td>
        <td>Numero 2 </td>
        <td><input name="txtb2" type="text" id="txtb2" value="<?=$b[1]?>"
size="5" /></td>
    </tr>

    <tr>
        <td>Numero 3 </td>
        <td><input name="txta3" type="text" id="txta3" value="<?=$a[2]?>"
size="5" /></td>
        <td>Numero 3 </td>
        <td><input name="txtb3" type="text" id="txtb3" value="<?=$b[2]?>"
size="5" /></td>
    </tr>
    <tr>
        <td>Numero 4 </td>
        <td><input name="txta4" type="text" id="txta4" value="<?=$a[3]?>"
size="5" /></td>
        <td>Numero 4 </td>
        <td><input name="txtb4" type="text" id="txtb4" value="<?=$b[3]?>"
size="5" /></td>
    </tr>

    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>

    <tr>
        <td>Cantidad</td>
        <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" size="5" /></td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 77

Enunciado: Ordene 4 números usando el método de ordenación por intercambio (burbuja).

Análisis: Para la solución de este problema se requiere que el usuario ingrese 4 números; luego, que el sistema devuelva los números ordenados.

Entrada

- 4 números (n[4])

Salida

- 4 números ordenados (n[4])

Diseño: Interfaz de usuario

Problema 77 - Windows Internet Explorer

http://localhost/Cap07/Problema77.php

Archivo Edición Ver Favoritos Herramientas Ayu

Problema 77

Numero 1

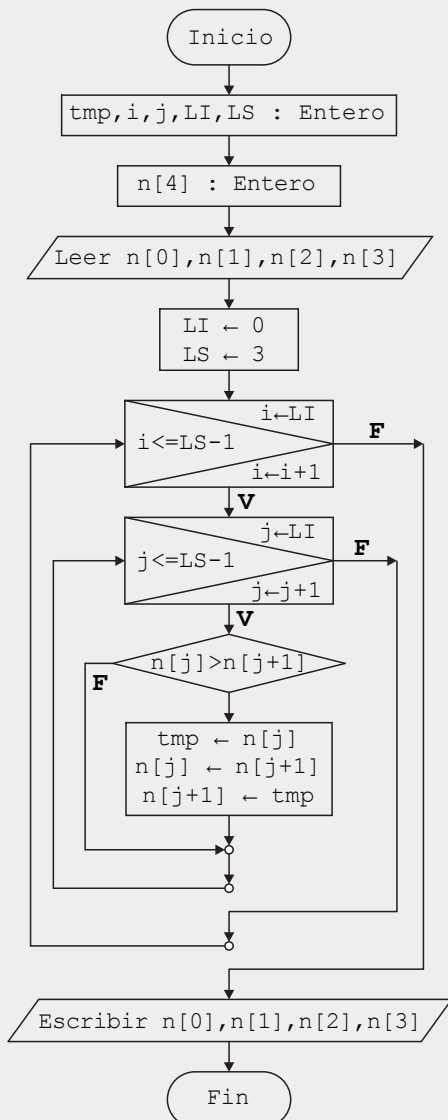
Numero 2

Numero 3

Numero 4

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

tmp, i, j, LI, LS : Entero

//Arreglos (Vector)

n[4] : Entero

//Entrada

Leer n[0], n[1], n[2], n[3]

//Proceso

LI ← 0

LS ← 3

Para i ← LI Hasta LS-1 Inc 1

 Para j ← LI Hasta LS-1 Inc 1

 Si n[j] > n[j+1] Entonces

 tmp ← n[j]

 n[j] ← n[j+1]

 n[j+1] ← tmp

 Fin Si

 Fin Para

Fin Para

//Salida

Escribir n[0], n[1], n[2], n[3]

Fin

Codificación:

```

<?php

//Variables
$tmp = 0; $i = 0; $j = 0;
$LI = 0; $LS = 0;

//Arreglos
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0] = (int)$_POST["txtn1"];
    $n[1] = (int)$_POST["txtn2"];
    $n[2] = (int)$_POST["txtn3"];
    $n[3] = (int)$_POST["txtn4"];

    //Proceso
    $LI = 0;
    $LS = count($n)-1;
    for($i = $LI; $i <= $LS - 1; $i++){
        for($j = $LI; $j <= $LS - 1; $j++){
            if($n[$j] > $n[$j + 1]){
                $tmp = $n[$j];
                $n[$j] = $n[$j + 1];
                $n[$j + 1] = $tmp;
            }
        }
    }
}

?>
<html>
<head>
<title>Problema 77</title>
</head>
<body>
<form method="post" action="Problema77.php">
    <table width="154" border="0">

        <tr>
            <td colspan="2"><strong>Problema 77</strong></td>
        </tr>

        <tr>
            <td width="76">Numero 1 </td>
            <td width="68">
                <input name="txtn1" type="text" id="txtn1" value="<?=$n[0]?>"
size="5" />
            </td>
        </tr>

        <tr>
            <td>Numero 2 </td>
            <td><input name="txtn2" type="text" id="txtn2" value="<?=$n[1]?>"
size="5" /></td>
        </tr>
    </table>

```

```

<tr>
  <td>Numero 3 </td>
  <td><input name="txttn3" type="text" id="txttn3" value="<?=$n[2]?>"
size="5" /></td>
</tr>
<tr>
  <td>Numero 4 </td>
  <td><input name="txttn4" type="text" id="txttn4" value="<?=$n[3]?>"
size="5" /></td>
</tr>

<tr>
  <td>&nbsp;</td>
  <td>
    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
  </td>
</tr>
</table>
</form>
</body>
</html>

```

Problema n.º 78

Enunciado: Ingrese 6 números en un arreglo de dos dimensiones (matriz) de 3x2 y obtenga la suma de los números ingresados.

Análisis: Para la solución de este problema se requiere que el usuario ingrese 6 números; luego, que el sistema devuelva la suma de los números.

Entrada

- 6 números ($n[3][2]$)

Salida

- Suma (s)

Diseño:

Interfaz de usuario

Problema 78 - Windows Internet Explorer

http://localhost/Cap07/Problema78.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 78

Problema 78
Arreglos 3X2

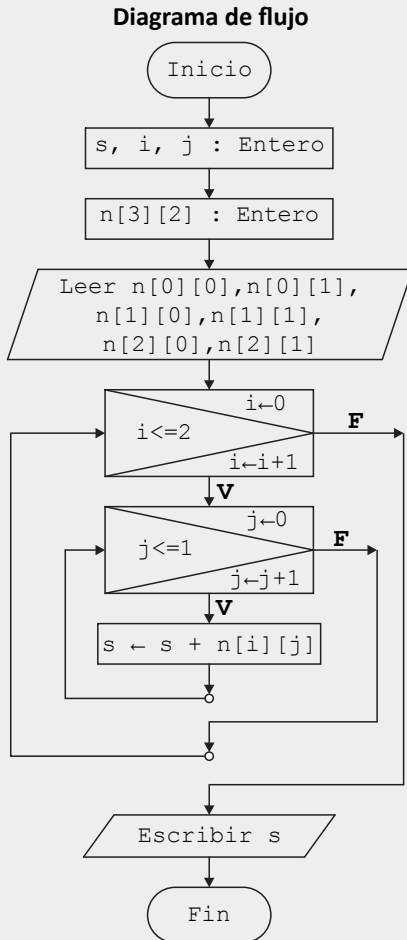
2	2
2	2
2	2

Suma 12

Calcular

Algoritmo

Pseudocódigo

**Inicio**

//Variables

s,i,j : Entero

//Arreglos (Matriz)

n[3][2] : Entero

//Entrada

Leer n[0][0],n[0][1],
n[1][0],n[0][1],
n[2][0],n[0][1],

//Proceso

Para i<-0 Hasta 2 Inc 1
 Para j<-0 Hasta 1 Inc 1
 s ← s + n[i][j]
 Fin Para
Fin Para

//Salida

Escribir s

Fin**Codificación:**

```

<?php
//Variables
$s = 0; $i = 0; $j = 0;

//Arreglos
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0][0] = (int)$_POST["txtn00"];
    $n[0][1] = (int)$_POST["txtn01"];
    $n[1][0] = (int)$_POST["txtn10"];
    $n[1][1] = (int)$_POST["txtn11"];
    $n[2][0] = (int)$_POST["txtn20"];
    $n[2][1] = (int)$_POST["txtn21"];

    //Proceso
    for($i = 0; $i <= 2; $i++)
        for($j = 0; $j <= 1; $j++)
            $s += $n[$i][$j];
}
?>
<html>
<head>
<title>Problema 78</title>
<style type="text/css">
<!--
  
```

```

.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema78.php">
    <table width="115" border="0">

        <tr>
            <td colspan="2"><strong>Problema 78</strong> </td>
        </tr>
        <tr>
            <td colspan="2"><strong>Arreglos 3X2</strong> </td>
        </tr>
        <tr>
            <td width="43"><input name="txtn00" type="text" id="txtn00"
value="<?=$n[0][0]?>" size="5" /> </td>
            <td width="62"><input name="txtn01" type="text" id="txtn01"
value="<?=$n[0][1]?>" size="5" /> </td>
        </tr>
        <tr>
            <td><input name="txtn10" type="text" id="txtn10" value="<?=$n[1]
[0]?>" size="5" /></td>
            <td><input name="txtn11" type="text" id="txtn11" value="<?=$n[1]
[1]?>" size="5" /></td>
        </tr>

        <tr>
            <td><input name="txtn20" type="text" id="txtn20" value="<?=$n[2]
[0]?>" size="5" /></td>
            <td><input name="txtn21" type="text" id="txtn21" value="<?=$n[2]
[1]?>" size="5" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>

        <tr>
            <td>Suma</td>
            <td><input name="txts" type="text" class="TextoFondo" id="txts"
value="<?=$s?>" size="5" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td><input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" /> </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 79

Enunciado: Ingrese 12 números en un arreglo bidimensional (matriz) de 4x3, y obtenga la suma de cada columna.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese 12 números; luego, que el sistema devuelva la suma de cada columna.

Entrada

- 12 números (n[4][3])

Salida

- Suma columna 1 (s0)
- Suma columna 2 (s1)
- Suma columna 3 (s2)

Diseño:
Interfaz de usuario

Problema 79 - Windows Internet Explorer

http://localhost/Cap07/Problema79.php

Archivo Edición Ver Favoritos Herramientas Ayuda

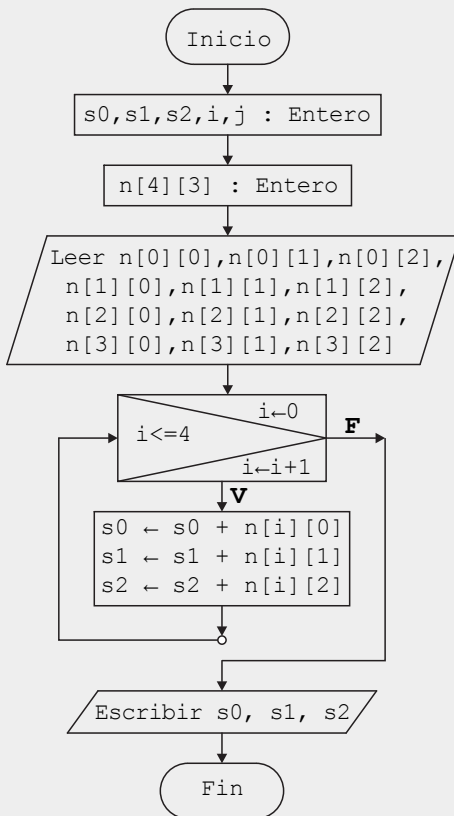
Problema 79

Problema 79
Arreglos 3X3

2	3	4
2	3	4
2	3	4

Suma

Diagrama de flujo



Algoritmo

Pseudocódigo

Inicio

//Variables

s0,s1,s2,i,j : Entero

//Arreglos (Matriz)

n[4][3] : Entero

//Entrada

Leer n[0][0],n[0][1],n[0][2],
n[1][0],n[1][1],n[1][2],
n[2][0],n[2][1],n[2][2],
n[3][0],n[3][1],n[3][2],

//Proceso

Para i←0 Hasta 4 Inc 1

s0 ← s0 + n[i][0]

s1 ← s1 + n[i][1]

s2 ← s2 + n[i][2]

Fin Para

//Salida

Escribir s0, s1, s2

Fin

Codificación:

```

<?php
//Variables
$s0 = 0; $s1 = 0; $s2 = 0; $i = 0; $j = 0;

//Arreglos
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0][0] = (int)$_POST["txtn00"];
    $n[0][1] = (int)$_POST["txtn01"];
    $n[0][2] = (int)$_POST["txtn02"];
    $n[1][0] = (int)$_POST["txtn10"];
    $n[1][1] = (int)$_POST["txtn11"];
    $n[1][2] = (int)$_POST["txtn12"];
    $n[2][0] = (int)$_POST["txtn20"];
    $n[2][1] = (int)$_POST["txtn21"];
    $n[2][2] = (int)$_POST["txtn22"];
    $n[3][0] = (int)$_POST["txtn30"];
    $n[3][1] = (int)$_POST["txtn31"];
    $n[3][2] = (int)$_POST["txtn32"];

    //Proceso
    for($i = 0; $i<=3; $i++) {
        $s0 += $n[$i][0];
        $s1 += $n[$i][1];
        $s2 += $n[$i][2];
    }
}
?>
<html>
<head>
<title>Problema 79</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema79.php">
    <table width="205" border="0">

        <tr>
            <td colspan="4"><strong>Problema 79 </strong></td>
        </tr>
        <tr>
            <td colspan="4"><strong>Arreglos 3X3 </strong></td>
        </tr>
        <tr>
            <td>
                <td width="52">&nbsp;</td>
                <td width="39"><input name="txtn00" type="text" id="txtn00"
value="<?=$n[0][0]?>" size="5" /> </td>
                <td width="40"><input name="txtn01" type="text" id="txtn01"
value="<?=$n[0][1]?>" size="5" /> </td>
                <td width="56">
                    <input name="txtn02" type="text" id="txtn02" value="<?=$n[0]
[2]?>" size="5" /> </td>
            </td>
        </tr>
    </table>
</form>

```

```

</tr>
<tr>
  <td>&nbsp;</td>
  <td><input name="txtn10" type="text" id="txtn10" value="<?=$n[1]
[0]?>" size="5" /></td>
  <td><input name="txtn11" type="text" id="txtn11" value="<?=$n[1]
[1]?>" size="5" /></td>
  <td><input name="txtn12" type="text" id="txtn12" value="<?=$n[1]
[2]?>" size="5" /></td>
</tr>
<tr>
  <td>&nbsp;</td>
  <td><input name="txtn20" type="text" id="txtn20" value="<?=$n[2]
[0]?>" size="5" /></td>
  <td><input name="txtn21" type="text" id="txtn21" value="<?=$n[2]
[1]?>" size="5" /></td>
  <td><input name="txtn22" type="text" id="txtn22" value="<?=$n[2]
[2]?>" size="5" /></td>
</tr>

<tr>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>

<tr>
  <td>Suma </td>
  <td><input name="txts0" type="text" class="TextoFondo" id="txts0"
value="<?=$s0?>" size="5" /></td>
  <td><input name="txts1" type="text" class="TextoFondo" id="txts1"
value="<?=$s1?>" size="5" /></td>
  <td><input name="txts2" type="text" class="TextoFondo" id="txts2"
value="<?=$s2?>" size="5" /></td>
</tr>

<tr>
  <td>&nbsp;</td>
  <td colspan="3"><input name="btnCalcular" type="submit"
id="btnCalcular" value="Calcular" />
</td>
</tr>
</table>
</form>
</body>
</html>

```

Problema n.º 80

Enunciado: Almacene en una matriz de 3 x 2, 6 números y obtenga la cantidad de pares e impares.

Análisis: Para la solución de este problema se requiere que el usuario ingrese 6 números; luego, que el sistema devuelva la cantidad de pares e impares.

Entrada

- 6 números ($n[3][2]$)

Salida

- Cantidad de pares (cp)
- Cantidad de impares (ci)

Diseño: Interfaz de usuario

Problema 80

Arreglos 3X2

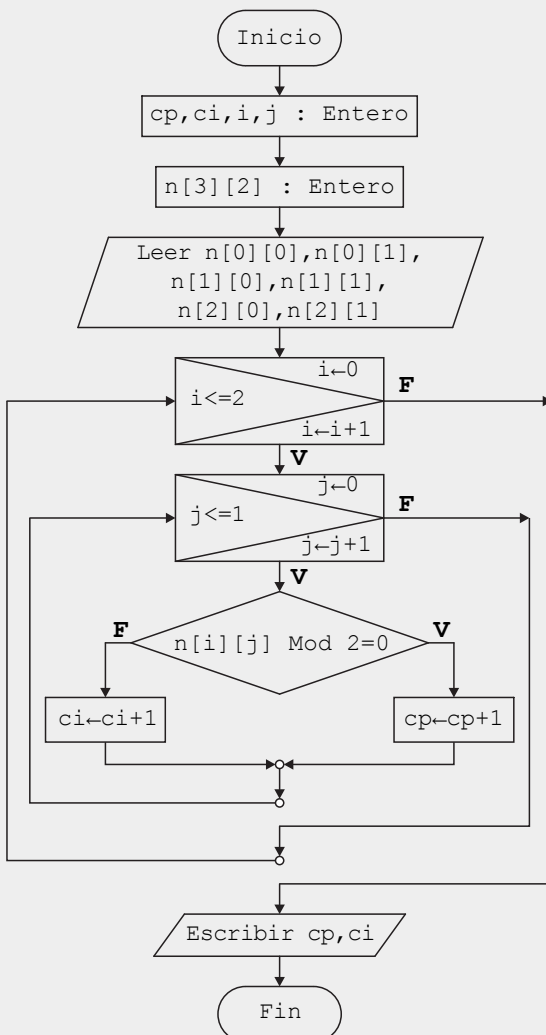
2	3
8	6
4	5

Cant. Pares: 4

Cant. Impares: 2

Calcular

Diagrama de flujo



Algoritmo

Pseudocódigo

Inicio

//Variables

cp, ci, i, j : Entero

//Arreglos (Matriz)

n[3][2] : Entero

//Entrada

Leer n[0][0], n[0][1],
n[1][0], n[1][1],
n[2][0], n[2][1]

//Proceso

```

Para i ← 0 Hasta 2 Inc 1
  Para j ← 0 Hasta 1 Inc 1
    Si n[i][j] Mod 2 = 0 Entonces
      cp ← cp + 1
    SiNo
      ci ← ci + 1
    Fin Si
  Fin Para
Fin Para
  
```

//Salida

Escribir cp, ci

Fin

Codificación:

```

<?php

//Variables
$cp = 0; $ci = 0; $i = 0; $j = 0;

//Arreglos
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0][0] = (int)$_POST["txtn00"];
    $n[0][1] = (int)$_POST["txtn01"];
    $n[1][0] = (int)$_POST["txtn10"];
    $n[1][1] = (int)$_POST["txtn11"];
    $n[2][0] = (int)$_POST["txtn20"];
    $n[2][1] = (int)$_POST["txtn21"];

    //Proceso
    for($i = 0; $i <= 2; $i++) {
        for($j = 0; $j <= 1; $j++) {
            if($n[$i][$j] % 2 == 0)
                $cp += 1;
            else
                $ci += 1;
        }
    }
}

?>
<html>
<head>
<title>Problema 80</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema80.php">
    <table width="238" border="0">

        <tr>
            <td colspan="3"><strong>Problema 80 </strong></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td colspan="2"><strong>Arreglos 3X2 </strong></td>
        </tr>
        <tr>
            <td width="117">&nbsp;</td>
            <td width="33"><input name="txtn00" type="text" id="txtn00"
value="<?=$n[0][0]?>" size="5" /> </td>

```

```

        <td width="51">
            <input name="txtn01" type="text" id="txtn01" value="<?=$n[0]
[1]?>" size="5" />        </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td><input name="txtn10" type="text" id="txtn10" value="<?=$n[1]
[0]?>" size="5" /></td>
            <td><input name="txtn11" type="text" id="txtn11" value="<?=$n[1]
[1]?>" size="5" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td><input name="txtn20" type="text" id="txtn20" value="<?=$n[2]
[0]?>" size="5" /></td>
            <td><input name="txtn21" type="text" id="txtn21" value="<?=$n[2]
[1]?>" size="5" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>

        <tr>
            <td>Cant. Pares </td>
            <td><input name="txtcp" type="text" class="TextoFondo" id="txtcp"
value="<?=$cp?>" size="5" /></td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Cant. Impares </td>
            <td><input name="txtci" type="text" class="TextoFondo" id="txtci"
value="<?=$ci?>" size="5" /></td>
            <td>&nbsp;</td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td colspan="2"><input name="btnCalcular" type="submit"
id="btnCalcular" value="Calcular" />        </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 81

Enunciado: Busque un número dentro de una matriz de 4x3 y determine la posición, y si existe o no el número buscado. Use el método de búsqueda secuencial.

Análisis: Para la solución de este problema se requiere que el usuario ingrese dos números; luego, que el sistema devuelva la cantidad de números positivos y negativos.

Entrada

- Matriz ($n[4][3]$)
- Número a buscar (nb)

Salida

- Respuesta (r)
- Posición 1.ra dim. (p1)
- Posición 2.da dim. (p2)

Diseño:**Interfaz de usuario**

Problema 81

Arreglos 3X3

5	3	6
3	1	7
9	8	0

Num. a buscar

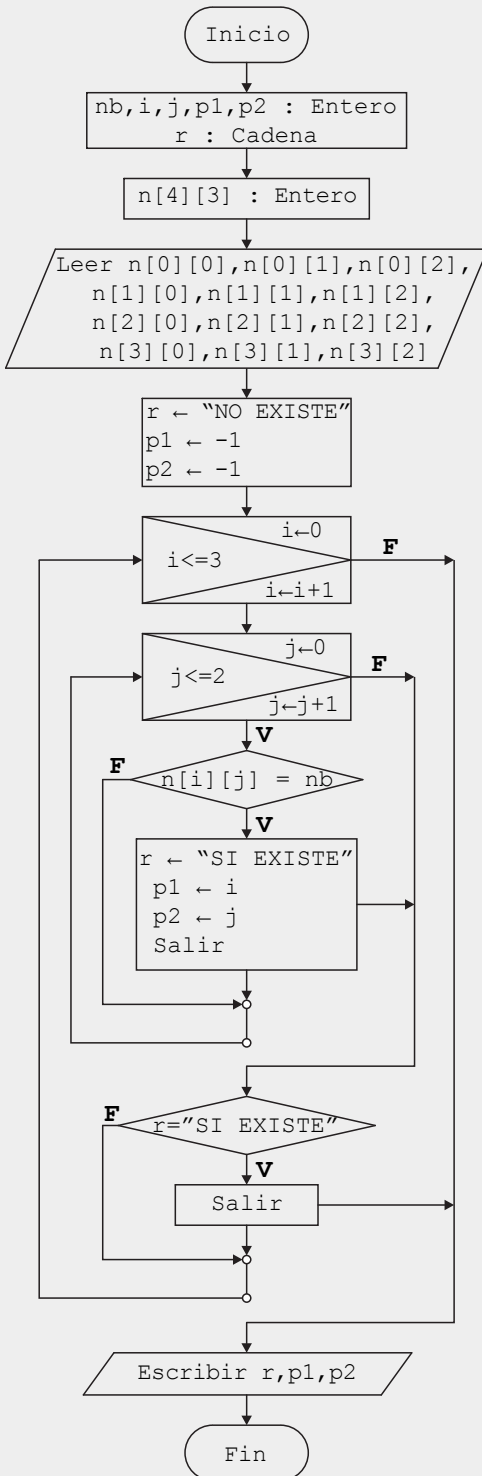
Respuesta

Posicion 1era Dim.

Posicion 2da Dim.

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

nb, i, j, p1, p2 : Entero
r : Cadena

//Arreglos (Matriz)

n[4][3] : Entero

//Entrada

Leer n[0][0], n[0][1], n[0][2],
n[1][0], n[1][1], n[1][2],
n[2][0], n[2][1], n[2][2],
n[3][0], n[3][1], n[3][2]

//Proceso

r ← "NO EXISTE"
p1 ← -1
p2 ← -1
Para i ← 0 Hasta 3 Inc 1
 Para j ← 0 Hasta 2 Inc 1
 Si n[i][j] = nb Entonces
 r ← "SI EXISTE"
 p1 ← i
 p2 ← j
 Salir
 Fin Si
 Fin Para
 Si r = "SI EXISTE" Entonces
 Salir
 Fin Si
Fin Para

//Salida

Escribir r, p1, p2

Fin

Codificación:

```

<?php

//Variables
$nb = 0; $i = 0; $j = 0; $p1 = 0; $p2 = 0;
$r = "";

//Arreglos
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0][0] = (int)$_POST["txtn00"];
    $n[0][1] = (int)$_POST["txtn01"];
    $n[0][2] = (int)$_POST["txtn02"];
    $n[1][0] = (int)$_POST["txtn10"];
    $n[1][1] = (int)$_POST["txtn11"];
    $n[1][2] = (int)$_POST["txtn12"];
    $n[2][0] = (int)$_POST["txtn20"];
    $n[2][1] = (int)$_POST["txtn21"];
    $n[2][2] = (int)$_POST["txtn22"];
    $n[3][0] = (int)$_POST["txtn30"];
    $n[3][1] = (int)$_POST["txtn31"];
    $n[3][2] = (int)$_POST["txtn32"];
    $nb = (int)$_POST["txtnb"];

    //Proceso
    $r = "NO EXISTE";
    $p1 = -1;
    $p2 = -1;

    for($i = 0; $i<=3;$i++){
        for($j = 0;$j<=2;$j++){
            if($n[$i][$j] == $nb){
                $r = "SI EXISTE";
                $p1 = $i;
                $p2 = $j;
                break;
            }
        }
        if($r == "SI EXISTE")
            break;
    }
}

?>
<html>
<head>
<title>Problema 81</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>

```

```

<form method="post" action="Problema81.php">
  <table width="311" border="0">

    <tr>
      <td colspan="4"><strong>Problema 81 </strong></td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td colspan="3"><strong>Arreglos 3X3</strong> </td>
    </tr>
    <tr>
      <td width="131">&nbsp;</td>
      <td width="35"><input name="txtn00" type="text" id="txtn00"
value="<?=$n[0][0]?" size="5" /> </td>
      <td width="35"><input name="txtn01" type="text" id="txtn01"
value="<?=$n[0][1]?" size="5" /> </td>
      <td width="46">
        <input name="txtn02" type="text" id="txtn02" value="<?=$n[0]
[2]?" size="5" /> </td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td><input name="txtn10" type="text" id="txtn10" value="<?=$n[0]
[1]?" size="5" /></td>
      <td><input name="txtn11" type="text" id="txtn11" value="<?=$n[1]
[1]?" size="5" /></td>
      <td><input name="txtn12" type="text" id="txtn12" value="<?=$n[1]
[2]?" size="5" /></td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td><input name="txtn20" type="text" id="txtn20" value="<?=$n[2]
[0]?" size="5" /></td>
      <td><input name="txtn21" type="text" id="txtn21" value="<?=$n[2]
[1]?" size="5" /></td>
      <td><input name="txtn22" type="text" id="txtn22" value="<?=$n[2]
[2]?" size="5" /></td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td><input name="txtnb" type="text" id="txtnb" value="<?=$nb?"
size="5" /></td>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>

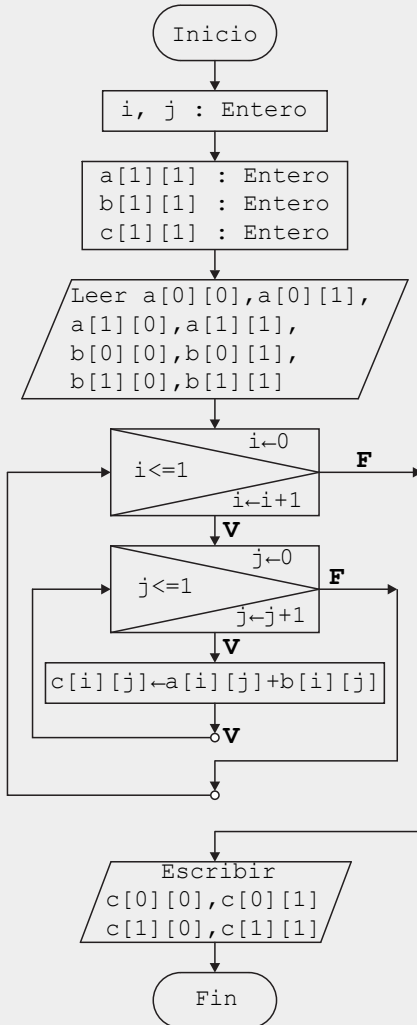
    <tr>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>

    <tr>
      <td colspan="4"><strong>Respuesta</td>
      <td colspan="3"><input name="txtr" type="text" class="TextoFondo"
id="txtr" value="<?=$r?" size="15" /></td>
    </tr>

```


Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

i, j : Entero

//Arreglos (Matriz)

a[1][1] : Entero

b[1][1] : Entero

c[1][1] : Entero

//Entrada

Leer a[0][0],a[0][1],

a[1][0],a[1][1],

b[0][0],b[0][1],

b[1][0],b[1][1]

//Proceso

Para i←0 Hasta 1 Inc 1

Para j←0 Hasta 1 Inc 1

c[i][j]←a[i][j]+b[i][j]

Fin Para

Fin Para

//Salida

Escribir c[0][0],c[0][1],

c[1][0],c[1][1]

Fin

Codificación:

```

<?php

//Variables
$i = 0; $j = 0;

//Arreglos
$a = array();
$b = array();
$c = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $a[0][0] = (int)$_POST["txta00"];
    $a[0][1] = (int)$_POST["txta01"];
    $a[1][0] = (int)$_POST["txta10"];
    $a[1][1] = (int)$_POST["txta11"];
}
  
```

```

$b[0][0] = (int)$ _POST["txtb00"];
$b[0][1] = (int)$ _POST["txtb01"];
$b[1][0] = (int)$ _POST["txtb10"];
$b[1][1] = (int)$ _POST["txtb11"];

//Proceso
for($i = 0; $i<=1; $i++){
    for($j = 0; $j<=1; $j++){
        $c[$i][$j] = $a[$i][$j] + $b[$i][$j];
    }
}

?>
<html>
<head>
<title>Problema 82</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema82.php">
    <table width="279" border="0">

        <tr>
            <td colspan="2"><strong>Problema 82</strong> </td>
            <td colspan="2">&nbsp;</td>
        </tr>
        <tr>
            <td colspan="2"><strong>Arreglo A</strong> </td>
            <td colspan="2"><strong>Arreglo B </strong> </td>
        </tr>
        <tr>
            <td width="35"><input name="txta00" type="text" id="txta00"
value="<?=$a[0][0]?>" size="5" /></td>
            <td width="157"><input name="txta01" type="text" id="txta01"
value="<?=$a[0][1]?>" size="5" /></td>
            <td width="34"><input name="txtb00" type="text" id="txtb00"
value="<?=$b[0][0]?>" size="5" /></td>
            <td width="35"><input name="txtb01" type="text" id="txtb01"
value="<?=$b[0][1]?>" size="5" /></td>
        </tr>
        <tr>
            <td><input name="txta10" type="text" id="txta10" value="<?=$a[1]
[0]?>" size="5" /></td>
            <td><input name="txta11" type="text" id="txta11" value="<?=$a[1]
[1]?>" size="5" /></td>
            <td><input name="txtb10" type="text" id="txtb10" value="<?=$b[1]
[0]?>" size="5" /></td>
            <td><input name="txtb11" type="text" id="txtb11" value="<?=$b[1]
[1]?>" size="5" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
    </table>

```

```

<tr>
  <td colspan="4"><strong>Arreglo C </strong> </td>
</tr>
<tr>
  <td><input name="txtc00" type="text" class="TextoFondo" id="txtc00"
value="<?=$c[0][0]?>" size="5" /></td>
  <td><input name="txtc01" type="text" class="TextoFondo" id="txtc01"
value="<?=$c[0][1]?>" size="5" /></td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td><input name="txtc10" type="text" class="TextoFondo" id="txtc10"
value="<?=$c[1][0]?>" size="5" /></td>
  <td><input name="txtc11" type="text" class="TextoFondo" id="txtc11"
value="<?=$c[1][1]?>" size="5" /></td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td colspan="4"><input name="btnCalcular" type="submit"
id="btnCalcular" value="Calcular" /></td>
</tr>
</table>
</form>
</body>
</html>

```

Problema n.º 83

Enunciado: Ingrese 6 números en una matriz de 3x2 y obtenga el número mayor ingresado.

Análisis: Para la solución de este problema se requiere que el usuario ingrese 6 números; luego, que el sistema devuelva el número mayor.

Entrada

- 6 números ($n[3][2]$)

Salida

- Mayor (m)

Diseño:

Interfaz de usuario

Problema 83 - Windows Internet Explorer

http://localhost/Cap07/Problema83.php

Archivo Edición Ver Favoritos Herramientas Ayu

Problema 83

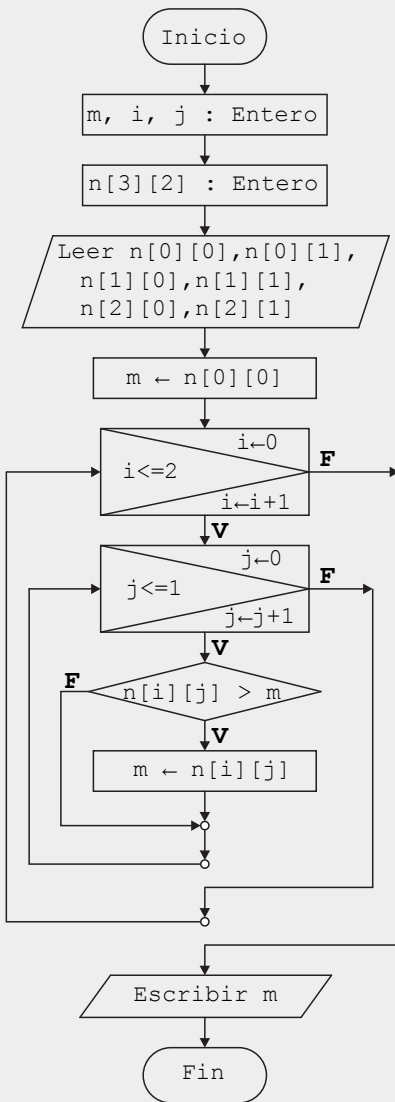
Problema 83

Arreglo 3X2

2	7
4	8
6	3

Mayor 8

Calcular

Diagrama de flujo**Algoritmo****Pseudocódigo****Inicio****//Variables**

m, i, j : Entero

//Arreglos (Matriz)

n[3][2] : Entero

//Entrada

Leer n[0][0], n[0][1],
n[1][0], n[1][1],
n[2][0], n[2][1]

//Proceso

m ← n[0][0]
Para i ← 0 Hasta 2 Inc 1
 Para j ← 0 Hasta 1 Inc 1
 Si n[i][j] > m Entonces
 m ← n[i][j]
 Fin Si
 Fin Para
Fin Para

//Salida

Escribir m

Fin

Codificación:

```

<?php

//Variables
$m = 0; $i = 0; $j = 0;

//Arreglos
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0][0] = (int)$_POST["txtn00"];
    $n[0][1] = (int)$_POST["txtn01"];
    $n[1][0] = (int)$_POST["txtn10"];
    $n[1][1] = (int)$_POST["txtn11"];
    $n[2][0] = (int)$_POST["txtn20"];
    $n[2][1] = (int)$_POST["txtn21"];

    //Proceso
    $m = $n[0][0];
    for($i = 0; $i <= 2; $i++) {
        for($j = 0; $j <= 1; $j++) {
            if($n[$i][$j] > $m)
                $m = $n[$i][$j];
        }
    }
}

?>
<html>
<head>
<title>Problema 83</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema83.php">
    <table width="113" border="0">
        <tr>
            <td colspan="2"><strong>Problema 83 </strong></td>
        </tr>
        <tr>
            <td colspan="2"><strong>Arreglo 3X2</strong> </td>
        </tr>
        <tr>
            <td width="41"><input name="txtn00" type="text" id="txtn00"
value="<?=$n[0][0]?" size="5" /></td>
            <td width="232"><input name="txtn01" type="text" id="txtn01"
value="<?=$n[0][1]?" size="5" /></td>
        </tr>
        <tr>
            <td><input name="txtn10" type="text" id="txtn10" value="<?=$n[1]
[0]?" size="5" /></td>
            <td><input name="txtn11" type="text" id="txtn11" value="<?=$n[1]
[1]?" size="5" /></td>
        </tr>
    </table>

```

```

        <tr>
            <td><input name="txtn20" type="text" id="txtn20" value="<?=$n[2]
[0]?>" size="5" /></td>
            <td><input name="txtn21" type="text" id="txtn21" value="<?=$n[2]
[1]?>" size="5" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Mayor</td>
            <td><input name="txtm" type="text" class="TextoFondo" id="txtm"
value="<?=$m?>" size="5" /> </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 84

Enunciado: Ingrese 6 números en una matriz de 3x2 y ordene los números de cada columna.

Análisis: Para la solución de este problema se requiere que el usuario ingrese 6 números; luego, que el sistema devuelva las columnas ordenadas.

Entrada

- 6 números ($n[3][2]$)

Salida

- Cada columna ordenada ($n[3][2]$)

Diseño:

Interfaz de usuario

Problema 84 - Windows Internet Explorer

http://localhost/Cap07/Problema84.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 84

Problema 84

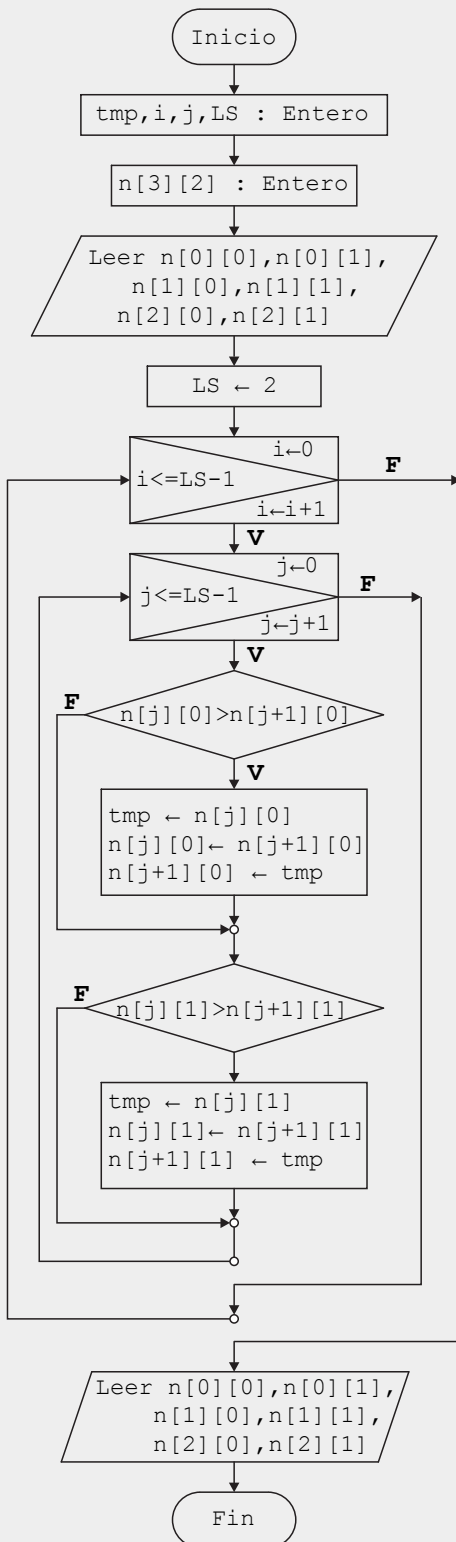
Ordenado por columnas

9	8
4	2
7	8

Calcular

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

tmp, i, j, LS : Entero

//Arreglos (Matriz)

n[3][2] : Entero

//Entrada

Leer n[0][0], n[0][1],
n[1][0], n[1][1],
n[2][0], n[2][1]

//Proceso

LS ← 2

Para i ← 0 Hasta LS-1 Inc 1

Para j ← 0 Hasta LS-1 Inc 1

Si n[j][0] > n[j+1][0] Entonces

tmp ← n[j][0]

n[j][0] ← n[j+1][0]

n[j+1][0] ← tmp

Fin Si

Si n[j][1] > n[j+1][1] Entonces

tmp ← n[j][1]

n[j][1] ← n[j+1][1]

n[j+1][1] ← tmp

Fin Si

Fin Para

Fin Para

//Salida

Escribir n[0][0], n[0][1],
n[1][0], n[1][1],
n[2][0], n[2][1]

Fin

Codificación:

```

<?php

//Variables
$tmp = 0; $i = 0; $j = 0; $LS = 0;

//Arreglos
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0][0] = (int)$_POST["txtn00"];
    $n[0][1] = (int)$_POST["txtn01"];
    $n[1][0] = (int)$_POST["txtn10"];
    $n[1][1] = (int)$_POST["txtn11"];
    $n[2][0] = (int)$_POST["txtn20"];
    $n[2][1] = (int)$_POST["txtn21"];

    //Proceso
    $LS = 2;
    for($i = 0; $i <= $LS - 1; $i++){
        for($j = 0; $j <= $LS - 1; $j++){
            if($n[$j][0] > $n[$j + 1][0]){
                $tmp = $n[$j][0];
                $n[$j][0] = $n[$j + 1][0];
                $n[$j + 1][0] = $tmp;
            }
            if($n[$j][1] > $n[$j + 1][1]){
                $tmp = $n[$j][1];
                $n[$j][1] = $n[$j + 1][1];
                $n[$j + 1][1] = $tmp;
            }
        }
    }
}
?>
<html>
<head>
<title>Problema 84</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema84.php">
    <table width="172" border="0">
        <tr>
            <td colspan="2"><strong>Problema 84</strong> </td>
        </tr>
        <tr>
            <td colspan="2"><strong>Ordenado por columnas</strong> </td>
        </tr>
        <tr>
            <td width="33"><input name="txtn00" type="text" id="txtn00"
value="<?=$n[0][0]?>" size="5" /></td>
            <td width="129"><input name="txtn01" type="text" id="txtn01"
value="<?=$n[0][1]?>" size="5" /></td>
        </tr>
    </table>

```

```

        <td><input name="txtn10" type="text" id="txtn10" value="<?=$n[1]
[0]?>" size="5" /></td>
        <td><input name="txtn11" type="text" id="txtn11" value="<?=$n[1]
[1]?>" size="5" /></td>
    </tr>
    <tr>
        <td><input name="txtn20" type="text" id="txtn20" value="<?=$n[2]
[0]?>" size="5" /></td>
        <td><input name="txtn21" type="text" id="txtn21" value="<?=$n[2]
[1]?>" size="5" /></td>
    </tr>
    <tr>
        <td colspan="2">
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 85

Enunciado: Almacene 9 números en una matriz de 3x3 y obtenga los números ordenados.

Análisis: Para la solución de este problema se requiere que el usuario ingrese 9 números; luego, que el sistema devuelva la matriz con los números ordenados.

Entrada

- 9 números ($n[3][3]$)

Salida

- 9 números ordenados ($n[3][3]$)

Diseño:

Interfaz de usuario

Problema 85 - Windows Internet Explorer

http://localhost/Cap07/Problema85.php

Archivo Edición Ver Favoritos Herramientas Ay

Problema 85

Problema 85
Arreglos 3X3

1	2	3
4	5	6
7	8	9

Ordenado

Numero 1

Numero 2

Numero 3

Numero 4

Numero 5

Numero 6

Numero 7

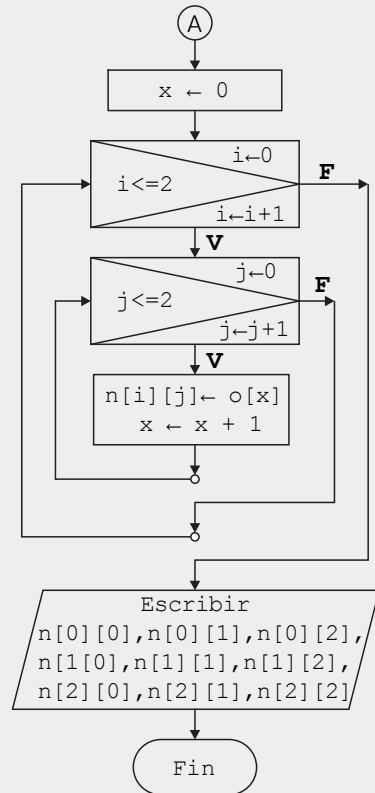
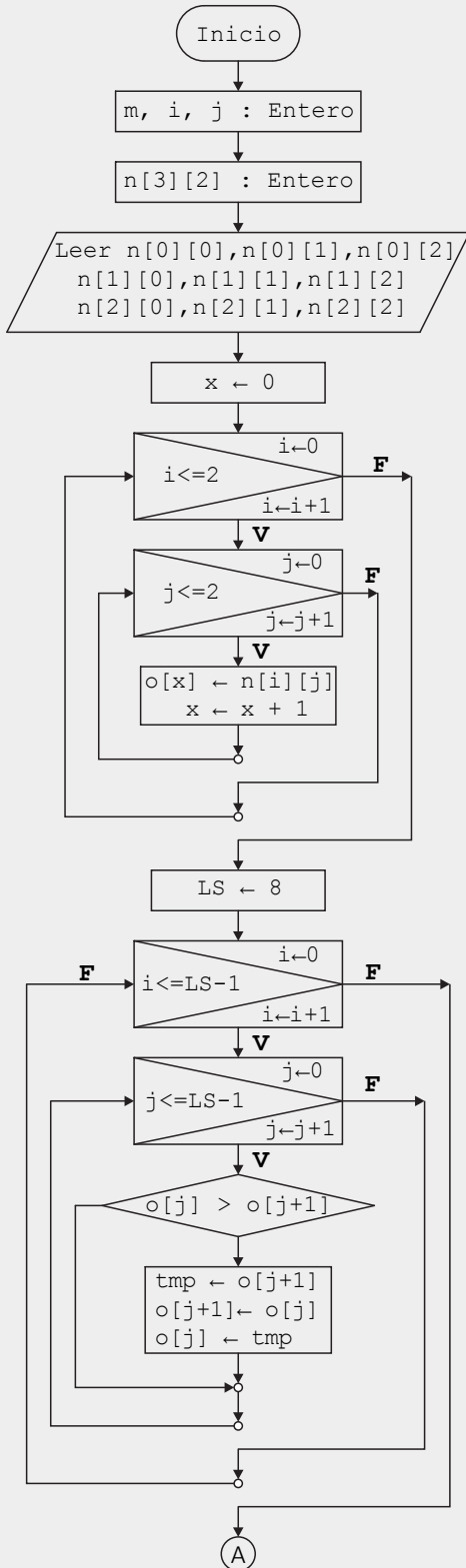
Numero 8

Numero 9

Calcular

Algoritmo

Diagrama de flujo



Pseudocódigo**Inicio****//Variables**

```
tmp, i, j, x, LS : Entero
flag : Logico
```

//Arreglos (Matriz y Vector)

```
n[3][3] : Entero
O[9] : Entero
```

//Entrada

```
Leer n[0][0],n[0][1],n[0][2],
      n[1][0],n[1][1],n[1][2],
      n[2][0],n[2][1],n[2][2]
```

//Proceso

```
x ← 0
Para i←0 Hasta 2 Inc 1
  Para j←0 Hasta 2 Inc 1
    o[x] ← n[i][j]
    x ← x + 1
  Fin Para
Fin Para

LS ← 8
Para i←0 Hasta LS-1 Inc 1
  Para j←0 Hasta LS-1 Inc 1
    Si o[j] > o[j+1] Entonces
      tmp ← o(j + 1)
      o(j + 1) ← o(j)
      o(j) ← tmp
    Fin Si
  Fin Para
Fin Para
```

```
x ← 0
Para i←0 Hasta 2 Inc 1
  Para j←0 Hasta 2 Inc 1
    n[i][j] ← o[x]
    x ← x + 1
  Fin Para
Fin Para
```

//Salida

```
Escribir n[0][0],n[0][1],n[0][2],
          n[1][0],n[1][1],n[1][2],
          n[2][0],n[2][1],n[2][2]
```

Fin

Codificación:

```

<?php

//Variables
$tmp = 0; $i = 0; $j = 0; $x = 0; $LS = 0;

//Arreglos
$n = array();
$o = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0][0] = (int)$_POST["txtn00"];
    $n[0][1] = (int)$_POST["txtn01"];
    $n[0][2] = (int)$_POST["txtn02"];
    $n[1][0] = (int)$_POST["txtn10"];
    $n[1][1] = (int)$_POST["txtn11"];
    $n[1][2] = (int)$_POST["txtn12"];
    $n[2][0] = (int)$_POST["txtn20"];
    $n[2][1] = (int)$_POST["txtn21"];
    $n[2][2] = (int)$_POST["txtn22"];

    //Proceso
    $x = 0;
    for($i = 0; $i<=2; $i++) {
        for($j = 0; $j<=2; $j++){
            $o[$x] = $n[$i][$j];
            $x++;
        }
    }

    $LS = count($o) - 1;
    for($i = 0; $i<=$LS - 1; $i++){
        for($j = 0; $j <= $LS - 1; $j++) {
            if($o[$j] > $o[$j + 1]) {
                $tmp = $o[$j + 1];
                $o[$j + 1] = $o[$j];
                $o[$j] = $tmp;
            }
        }
    }

    $x = 0;
    for($i = 0; $i<=2; $i++) {
        for($j = 0; $j<=2; $j++){
            $n[$i][$j] = $o[$x];
            $x++;
        }
    }
}
?>
<html>
<head>
<title>Problema 85</title>
<style type="text/css">

```

```

<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema85.php">
    <table width="218" border="0">

        <tr>
            <td colspan="4"><strong>Problema 85 </strong></td>
        </tr>
        <tr>
            <td colspan="4"><strong>Arreglos 3X3</strong> </td>
        </tr>
        <tr>
            <td width="87">&nbsp;</td>
            <td width="35"><input name="txtn00" type="text" id="txtn00"
value="<?=$n[0][0]?" size="5" /></td>
            <td width="32"><input name="txtn01" type="text" id="txtn01"
value="<?=$n[0][1]?" size="5" /></td>
            <td width="46"><input name="txtn02" type="text" id="txtn02"
value="<?=$n[0][2]?" size="5" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td><input name="txtn10" type="text" id="txtn10" value="<?=$n[1]
[0]?" size="5" /></td>
            <td><input name="txtn11" type="text" id="txtn11" value="<?=$n[1]
[1]?" size="5" /></td>
            <td><input name="txtn12" type="text" id="txtn12" value="<?=$n[1]
[2]?" size="5" /></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td><input name="txtn20" type="text" id="txtn20" value="<?=$n[2]
[0]?" size="5" /></td>
            <td><input name="txtn21" type="text" id="txtn21" value="<?=$n[2]
[1]?" size="5" /></td>
            <td><input name="txtn22" type="text" id="txtn22" value="<?=$n[2]
[2]?" size="5" /></td>
        </tr>
        <tr>
            <td><strong>Ordenado</strong></td>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Numero 1 </td>
            <td><input name="txto0" type="text" class="TextoFondo" id="txto0"
value="<?=$o[0]?" size="5" /> </td>
            <td>&nbsp;</td>
    </table>

```

```

        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>Numero 2 </td>
        <td><input name="txto1" type="text" class="TextoFondo" id="txto1"
value="<?=$o[1]?>" size="5" /></td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>Numero 3 </td>
        <td><input name="txto2" type="text" class="TextoFondo" id="txto2"
value="<?=$o[2]?>" size="5" /></td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>Numero 4 </td>
        <td><input name="txto3" type="text" class="TextoFondo" id="txto3"
value="<?=$o[3]?>" size="5" /></td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>Numero 5 </td>
        <td><input name="txto4" type="text" class="TextoFondo" id="txto4"
value="<?=$o[4]?>" size="5" /></td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>Numero 6 </td>
        <td><input name="txto5" type="text" class="TextoFondo" id="txto5"
value="<?=$o[5]?>" size="5" /></td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>Numero 7 </td>
        <td><input name="txto6" type="text" class="TextoFondo" id="txto6"
value="<?=$o[6]?>" size="5" /></td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>Numero 8 </td>
        <td><input name="txto7" type="text" class="TextoFondo" id="txto7"
value="<?=$o[7]?>" size="5" /></td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>Numero 9 </td>
        <td><input name="txto8" type="text" class="TextoFondo" id="txto8"
value="<?=$o[8]?>" size="5" /></td>

```

```
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>

    <tr>
        <td>&nbsp;</td>
        <td colspan="3">
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>
```

7.6 Problemas propuestos

Los siguientes problemas le servirán para medir su aprendizaje, es importante que los resuelva.

Propuesto n.º 51

Enunciado: Dados 4 números y almacénelo en un vector, luego obtenga la suma y el promedio de los valores almacenados.

Propuesto n.º 52

Enunciado: Dados 4 números, almacénelos en un vector, el número mayor y menor.

Propuesto n.º 53

Enunciado: Dados 6 números, almacénelos en un vector, luego obtenga cuántos números múltiplos de n ha ingresado.

Propuesto n.º 54

Enunciado: Ordene 5 números según la forma que se indique: «A» (ascendente) o «D» (descendente).

Propuesto n.º 55

Enunciado: Ingrese 6 números y determine cuántos números repetidos existen.

Propuesto n.º 56

Enunciado: Ingrese 6 números en una matriz de 3×2 y obtenga la suma de cada fila.

Propuesto n.º 57

Enunciado: Ingrese 6 números en una matriz de 3×2 y obtenga el promedio aritmético.

Propuesto n.º 58

Enunciado: En una matriz de 2×3 ingrese 6 números, multiplique su contenido por un valor K y obtenga la suma de los números de la matriz.

Propuesto n.º 59

Enunciado: Cree una matriz de «A» de 2×2 , otra «B» de 2×2 , y obtenga una matriz $C = A * B$.

Propuesto n.º 60

Enunciado: Cree una matriz de 4×3 y obtenga los números mayores de cada columna.

Cadenas de caracteres

8.1 Introducción

Inicialmente, las computadoras fueron creadas con la finalidad de resolver problemas aritméticos; sin embargo, hoy en día el manejo de datos alfanuméricos (texto) es importante y de gran utilidad para el procesamiento de operaciones con caracteres (cadenas). Una cadena de caracteres es una secuencia de cero o más símbolos, que incluye letras del alfabeto, dígitos y caracteres especiales.

8.2 Juego de caracteres

Los lenguajes de programación utilizan un conjunto de caracteres para comunicarse con las computadoras, dentro de las cuales existen diferentes tipos de juego de caracteres de los que destacan el ASCII, UNICODE, etc.

Standard ASCII (Caracteres alfanuméricos)

33	!	49	1	65	A	81	Q	97	a	113	q
34	“	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	‘	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	△
48	0	64	@	80	P	96	`	112	p		

Caracteres extendidos de ASCII

128	€	144	•	160		176	°	193	Á	209	Ñ	225	á	241	ñ
129	•	145	‘	161	¡	177	±	194	Â	210	Ò	226	â	242	ò
130	,	146	’	162	¢	178	²	195	Ã	211	Ó	227	ã	243	ó
131	f	147	“	163	£	179	³	196	Ä	212	Ô	228	ä	244	ô
132	„	148	”	164	¤	180	´	197	Å	213	Õ	229	å	245	õ
133	...	149	•	165	¥	181	µ	198	Æ	214	Ö	230	æ	246	ö
134	†	150	–	166	¦	182	¶	199	Ç	215	×	231	ç	247	÷
135	‡	151	—	167	§	183	·	200	È	216	Ø	232	è	248	ø
136	^	152	~	168	¨	184	,	201	É	217	Ù	233	é	249	ù
137	‰	153	™	169	©	185	¹	202	Ê	218	Ú	234	ê	250	ú
138	Š	154	š	170	ª	186	º	203	Ë	219	Û	235	ë	251	û
139	‹	156	œ	171	«	187	»	204	Ì	220	Ü	236	ì	252	ü
140	Œ	157	•	172	¬	188	¼	205	Í	221	Ý	237	í	253	ý
141	•	158	ž	173		189	½	206	Î	222	Þ	238	î	254	þ
142	Ž	159	Ÿ	174	®	190	¾	207	Ï	223	ß	239	ï	255	ÿ
143	•	192	À	175	–	191	¿	208	Ð	224	à	240	ä		

8.3 Carácter (*char*)

Representa un solo valor de tipo carácter, por lo general se representa con comillas simples.

Pseudocódigo

```
//Crear una variable character
c : Carácter

//Asignar un valor
c ← 'A'
```

PHP

```
//Crear una variable character
$c = '' ;

//Asignar un valor
$c = 'A' ;
```

8.4 Cadena de caracteres (*string*)

Representa un conjunto de caracteres y por lo general lo representamos entre comillas dobles.

Pseudocódigo

```
//Crear una variable cadena  
c : Cadena  
  
//Asignar un valor  
c ← "ABC"
```

PHP

```
//Cadena  
$c = "";  
  
//Asignar um valor  
$c = "ABC";
```

8.5 Operaciones con cadena

Para la manipulación de las cadenas, los lenguajes de programación incorporan una variedad de funciones y/o métodos que permiten realizar operaciones con cadenas.

Las operaciones con cadenas más usadas son:

- Concatenación
- Comparación
- Cálculo de longitud
- Extracción de cadenas (subcadenas)
- Búsqueda de cadenas
- Conversiones

8.6 Concatenación

Unir varias cadenas en una sola.

Pseudocódigo

```
//Unir cadenas  
c ← "ABC" + "XYZ"
```

PHP

```
//Unir cadenas  
$c = "ABC" + "XYZ";
```

8.7 Comparación

Igualdad y desigualdad de cadenas.

Pseudocódigo

```
//Igualdad (Falso)
"AAA" = "aaa"

//Desigualdad (Verdadero)
"LUISA" > "LUIS"
```

PHP

```
`Igualdad (Falso)
"AAA" == "aaa"

`Desigualdad (Verdadero)
"LUISA" > "LUIS"
```

8.8 Cálculo de longitud

Obtener la cantidad de caracteres de una cadena.

Pseudocódigo

```
//Retorna 3
l ← Longitud("aaa")
```

PHP

```
//Retorna 3
$l = strlen("aaa");
```

8.9 Extracción de cadenas (subcadenas)

Extraer una parte específica de la cadena; por lo general, cada carácter de una cadena se representa por una posición que inicia con 0. Es decir, «JUAN» consta de 4 caracteres, «J» es el primer carácter cuya posición es 0, «U» el segundo carácter y de posición 1, así sucesivamente.

En PHP las posiciones de los caracteres de una cadena inician con 1.

Pseudocódigo

```
//Extraer el primer caracter A
// 1 cantidad a extraer
c ← Izquierda("ABC",1)

//También se usa
// 0 posicion
// 1 cantidad a extraer
c ← subcadena("ABC",0,1)

//Extraer el último caracter C
// 1 cantidad a extraer
c ← Derecha("ABC",1)

//También se usa
// 2 posicion
// 1 cantidad a extraer
c ← subcadena("ABC",2,1)

//Extraer el segundo caracter B
// 2 posicion
// 1 cantidad a extraer
c ← Extraer("ABC",1,1)
c ← subcadena("ABC",1,1)
```

PHP

```
//Extraer el primer caracter A
$c = substr("ABC",1);

//Extraer el último caracter C
c = "ABC".substring(2,3);

//Extraer el segundo caracter B
$c = substr("ABC",1,1);
```

8.10 Búsqueda de cadenas

Buscar si una cadena se encuentra dentro de otra cadena más grande; si fuera así, la función devuelve la posición de la cadena encontrada; caso contrario, retorna FALSE.

Pseudocódigo

```
//Retorna 1  
p ← Indice("ABC", "B")
```

PHP

```
`Retorna 1  
$p = strpos("ABC", "B");
```

8.11 Conversiones

Convertir números a cadena o cadena de números a números es una tarea frecuente en los lenguajes de programación, así como convertir a mayúscula o minúscula una cadena de caracteres, u obtener el valor ASCII de un carácter, o devolver el carácter de un código ASCII.

Pseudocódigo

```
//Convertir a mayúscula (ABC)  
c ← Mayus("abc")  
  
//Convertir a minúscula (abc)  
c ← Minus("ABC")  
  
//Obtener el valor ASCII (A = 65)  
c ← Codigo("A")  
  
//Obtener el carácter de un código ASCII (65 = A)  
c ← Caracter(65)
```

PHP

```
//Convertir a mayúscula (ABC)  
$c = strtoupper("abc");  
  
//Convertir a minúscula (abc)  
$c = strtolower("ABC");  
  
//Obtener el valor ASCII (A = 65)  
$c = ord("A");  
  
//Obtener el carácter de un código ASCII (65 = A)  
$c = chr(65);
```

Problema n.º 86

Enunciado: Dado un nombre, obtener la cantidad de caracteres que contiene.

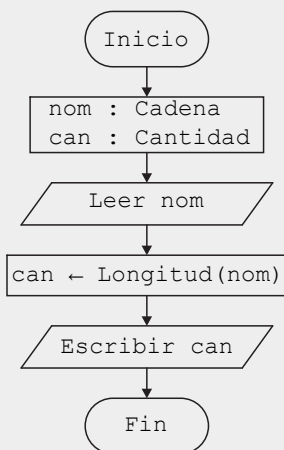
Análisis: Para la solución de este problema, se requiere que el usuario ingrese una cadena de caracteres; luego, que el sistema devuelva la cantidad de caracteres que contiene.

Entrada

- Cadena de caracteres (nom)

Salida

- Cantidad (can)

Diseño:**Interfaz de usuario**
Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio****//Variables**

nom : Cadena
can : Entero

//Entrada

Leer nom

//Proceso

can ← Longitud(nom)

//Salida

Escribir can

Fin

Codificación:

```

<?php

//Variables
$nom = ""; $can = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $nom = $_POST["txtnom"];

    //Proceso
    $can = strlen($nom);
}
?>
<html>
<head>
<title>Problema 86</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema86.php">
    <table width="268" border="0">

        <tr>
            <td colspan="2"><strong>Problema 86</strong> </td>
        </tr>
        <tr>
            <td width="86">Nombre</td>
            <td width="172">
                <input name="txtnom" type="text" id="txtnom" value="<?=$nom?>"
            </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Cantidad</td>
            <td><input name="txtcan" type="text" class="TextoFondo" id="txtcan"
value="<?=$can?>" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 87

Enunciado: Ingrese su nombre y apellido, y obtenga su nombre y apellido en mayúscula separado por una coma, así: XXXXX, XXXXX.

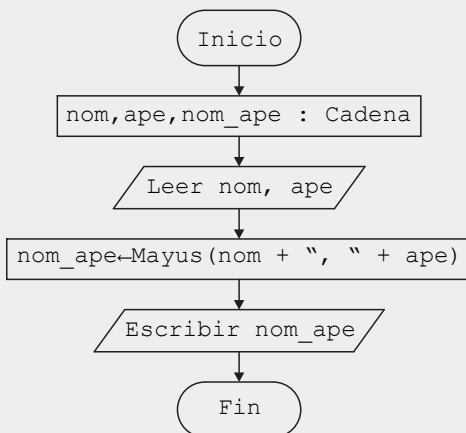
Análisis: Para la solución de este problema, se requiere que el usuario ingrese su nombre y apellido; luego, que el sistema devuelva su nombre y apellido separado por una coma y en mayúscula.

Entrada

- Nombre (nom)
- Apellido (ape)

Salida

- Nombre y apellido (nom_ape)

Diseño:**Interfaz de usuario**
Diagrama de flujo**Algoritmo****Pseudocódigo****Inicio****//Variables**

nom,ape,nom_ape : Cadena

//Entrada

Leer nom, ape

//Proceso

nom_ape←Mayus (nom + ", " + ape)

//Salida

Escribir nom_ape

Fin

Codificación:

```

<?php
//Variables
$nom = ""; $ape = ""; $nomape = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $nom = $_POST["txtnom"];
    $ape = $_POST["txtape"];

    //Proceso
    $nomape = strtoupper($nom . ", " . $ape);
}
?>
<html>
<head>
<title>Problema 87</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema87.php">
    <table width="268" border="0">

        <tr>
            <td colspan="2"><strong>Problema 87</strong> </td>
        </tr>
        <tr>
            <td width="86">Nombre</td>
            <td width="172">
                <input name="txtnom" type="text" id="txtnom" value="<?=$nom?>"
            </td>
        </tr>
        <tr>
            <td>Apellido </td>
            <td><input name="txtape" type="text" id="txtape" value="<?=$ape?>"
        </td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td colspan="2">Nmbre y Apellido </td>
        </tr>
        <tr>
            <td colspan="2"><input name="txtnomape" type="text" class="TextoFondo"
            id="txtnomape" value="<?=$nomape?>" size="40" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>

            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
                value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 88

Enunciado: Dado un carácter, devolver su código ASCII.

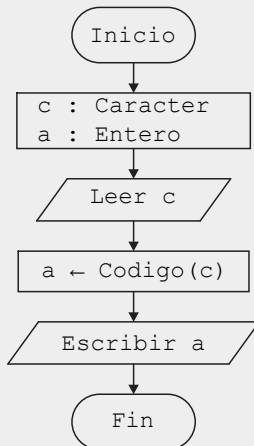
Análisis: Para la solución de este problema, se requiere que el usuario ingrese un carácter; luego, que el sistema devuelva el ASCII.

Entrada

- Carácter (c)

Salida

- ASCII (a)

Diseño:**Interfaz de usuario**
Diagrama de flujo**Algoritmo****Inicio**

//Variables
c : Carácter
a : Entero

//Entrada
Leer c

//Proceso
a ← Código(c)

//Salida
Escribir a

Fin**Pseudocódigo****Codificación:**

```

<?php

//Variables
$c = ""; $a = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $c = $_POST["txtc"];

```

```

        //Proceso
        $a = ord($c);
    }
    ?>
<html>
<head>
<title>Problema 88</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema88.php">
    <table width="268" border="0">

        <tr>
            <td colspan="2"><strong>Problema 88</strong> </td>
        </tr>
        <tr>
            <td width="86">Caracter</td>
            <td width="172">
                <input name="txtc" type="text" id="txtc" value="<?=$c?>" />
            </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>ASCII</td>
            <td><input name="txta" type="text" class="TextoFondo" id="txta"
value="<?=$a?>" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 89

Enunciado: Al ingresar una letra, determine si es una vocal.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese una letra; luego, que el sistema devuelva si es o no una vocal.

Entrada

- Letra (l)

Salida

- Respuesta (r)

Diseño: Interfaz de usuario

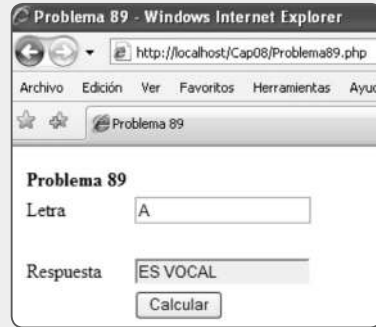
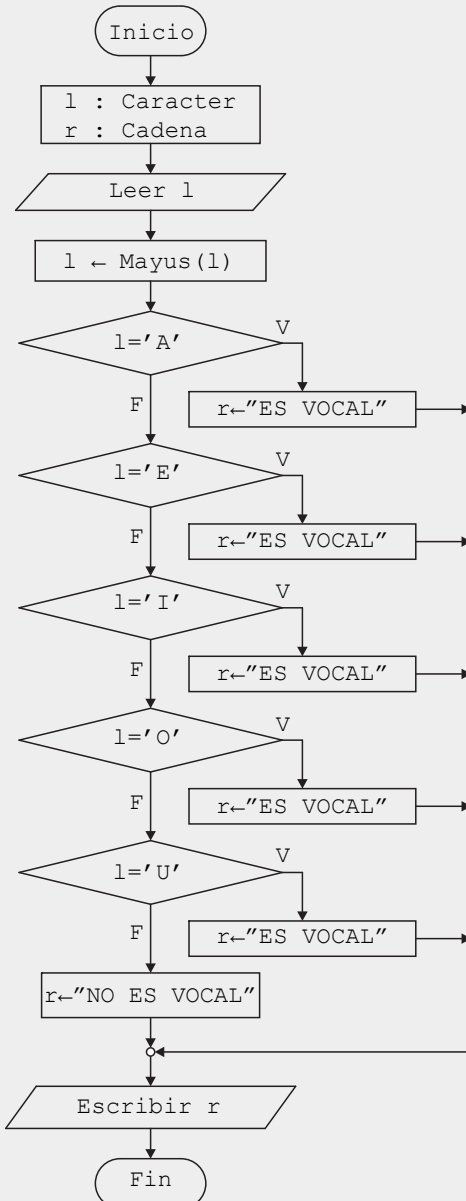


Diagrama de flujo



Algoritmo

Inicio

//Variables

l : Caracter

r : Cadena

//Entrada

Leer l

//Proceso

l ← Mayus(l)

Si l='A' Entonces

 R ← "ES VOCAL"

SiNoSi l='E' Entonces

 R ← "ES VOCAL"

SiNoSi l='I' Entonces

 R ← "ES VOCAL"

SiNoSi l='O' Entonces

 R ← "ES VOCAL"

SiNoSi l='U' Entonces

 R ← "ES VOCAL"

SiNo

 R ← "NO ES VOCAL"

Fin Si

//Salida

Escribir r

Fin

Codificación:

```

<?php
//Variables
$l = ""; $r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $l = $_POST["txt1"];

    //Proceso
    $l = strtoupper($l);
    if($l == "A")
        $r = "ES VOCAL";
    else if($l == "E")
        $r = "ES VOCAL";
    else if($l == "I")
        $r = "ES VOCAL";
    else if($l == "O")
        $r = "ES VOCAL";
    else if($l == "U")
        $r = "ES VOCAL";
    else
        $r = "NO ES VOCAL";
}
?>
<html>
<head>
<title>Problema 89</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema89.php">
    <table width="268" border="0">

        <tr>
            <td colspan="2"><strong>Problema 89</strong> </td>
        </tr>
        <tr>
            <td width="86">Letra</td>
            <td width="172">
                <input name="txt1" type="text" id="txt1" value="<?=$l?>" />
            </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Respuesta</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 90

Enunciado: Dado un carácter, determine si es una letra, número o símbolo.

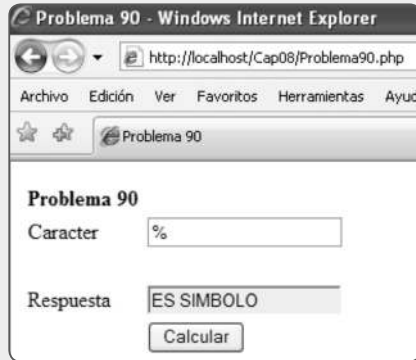
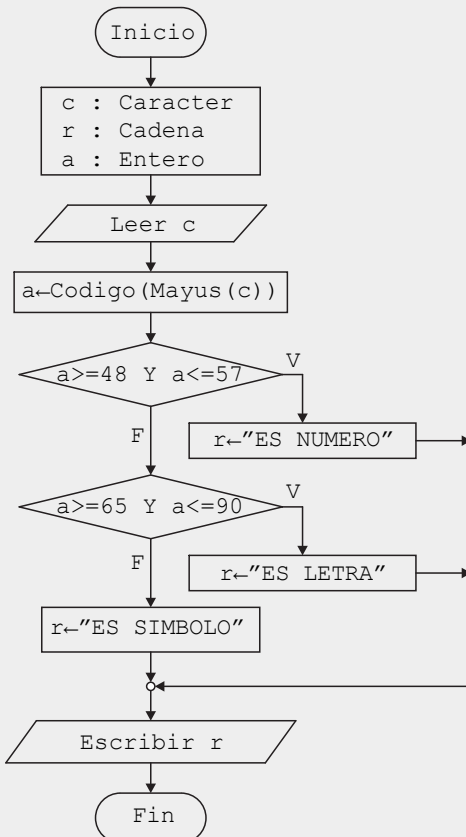
Análisis: Para la solución de este problema, se requiere que el usuario ingrese un carácter; luego, que el sistema devuelva si es letra, número o símbolo.

Entrada

- Carácter (c)

Salida

- Respuesta (r)

Diseño:**Interfaz de usuario****Diagrama de flujo****Algoritmo****Inicio****//Variables**

c : Carácter
r : Cadena
a : Entero

//Entrada

Leer c

//Proceso

```

a ← Codigo(Mayus(c))
Si a ≥ 48 Y a ≤ 57 Entonces
    r ← "ES NUMERO"
SiNo Si a ≥ 65 Y a ≤ 90 Entonces
    r ← "ES LETRA"
SiNo
    r ← "ES SIMBOLO"
Fin Si
  
```

//Salida

Escribir r

Fin

Codificación:

```

<?php

//Variables
$c = ""; $r = "";
$a = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $c = $_POST["txtc"];

    //Proceso
    $a = ord($c);
    if($a >= 48 && $a <= 57)
        $r = "ES NUMERO";
    else if($a >= 65 && $a <= 90)
        $r = "ES LETRA";
    else
        $r = "ES SIMBOLO";
}
?>
<html>
<head>
<title>Problema 90</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema90.php">
    <table width="268" border="0">

        <tr>
            <td colspan="2"><strong>Problema 90</strong> </td>
        </tr>
        <tr>
            <td width="86">Caracter</td>
            <td width="172">
                <input name="txtc" type="text" id="txtc" value="<?=$c?>" />
            </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Respuesta</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 91

Enunciado: Se desea obtener los N primeros caracteres de un nombre.

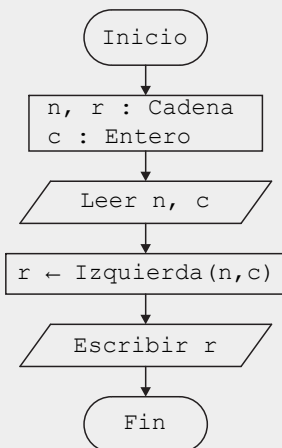
Análisis: Para la solución de este problema, se requiere que el usuario ingrese un nombre y una cantidad; luego, que el sistema devuelva los primeros caracteres indicados por la cantidad.

Entrada

- Nombre (n)
- Cantidad (c)

Salida

- Respuesta (r)

Diseño:**Interfaz de usuario**
Diagrama de flujo**Algoritmo****Inicio****//Variables**

n, r : Cadena
c : Entero

//Entrada

Leer n, c

//Proceso

r ← Izquierda(n,c)

//Salida

Escribir r

Fin

Codificación:

```

<?php

//Variables
$n = ""; $r = "";
$c = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = $_POST["txtn"];
    $c = $_POST["txtc"];

    //Proceso
    $r = substr($n, 0, $c);
}
?>
<html>
<head>
<title>Problema 91</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema91.php">
    <table width="268" border="0">

        <tr>
            <td colspan="2"><strong>Problema 91</strong> </td>
        </tr>
        <tr>
            <td width="86">Nombre</td>
            <td width="172">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>Cantidad</td>
            <td><input name="txtc" type="text" id="txtc" value="<?=$c?>" />
        </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Respuesta</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 92

Enunciado: Según las siguientes especificaciones, genere un código basado en el nombre ingresado.

Especificaciones para generar el código:

- 1.º carácter del código: Primer carácter del nombre.
- 2.º carácter del código: Tercer carácter del nombre.
- 3.º carácter del código: Último carácter del nombre.
- 4.º carácter del código: Cantidad de caracteres del nombre.

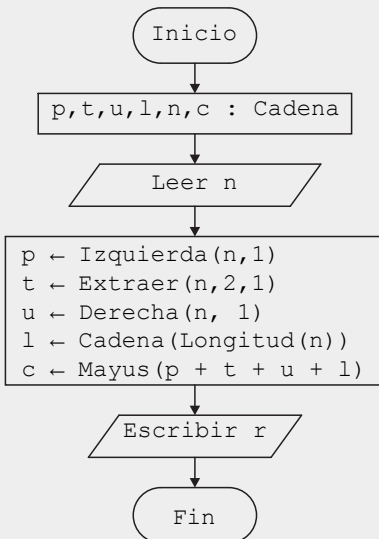
Análisis: Para la solución de este problema, se requiere que el usuario ingrese un nombre; luego, que el sistema procese y obtenga el código generado.

Entrada

- Nombre (n)

Salida

- Código (c)

Diseño:**Interfaz de usuario**
Algoritmo**Diagrama de flujo****Pseudocódigo****Inicio**

//Variables
p, t, u, l, n, c : Cadena

//Entrada
Leer n

//Proceso
p ← Izquierda(n, 1)
t ← Extraer(n, 2, 1)
u ← Derecha(n, 1)
l ← Cadena(Longitud(n))
c ← Mayus(p + t + u + l)

//Salida
Escribir c

Fin

Codificación:

```

<?php
//Variables
$p = ""; $t = ""; $u = ""; $l = "";
$n = ""; $c = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = $_POST["txtn"];

    //Proceso
    $p = substr($n,0,1);
    $t = substr($n,2,1);
    $u = substr($n,-1);
    $l = strlen($n);
    $c = strtoupper($p . $t . $u . $l);
}
?>
<html>
<head>
<title>Problema 92</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema92.php">
    <table width="268" border="0">

        <tr>
            <td colspan="2"><strong>Problema 92</strong> </td>
        </tr>
        <tr>
            <td width="86">Nombre</td>
            <td width="172">
                <input name="txtn" type="text" id="txtn" value="<?=$n?>" />
            </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Codigo</td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" /></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 93

Enunciado: Determine cuántas veces se repite una letra en una frase dada.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese una frase y una letra; luego, que el sistema devuelva la cantidad de veces que se repite la letra en la frase.

Entrada

- Frase (f)
- Letra (l)

Salida

- Cantidad (c)

Diseño:

Interfaz de usuario

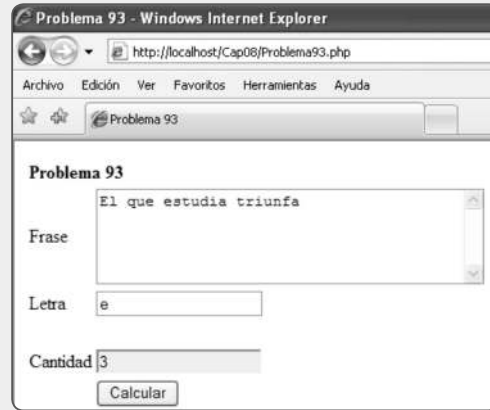
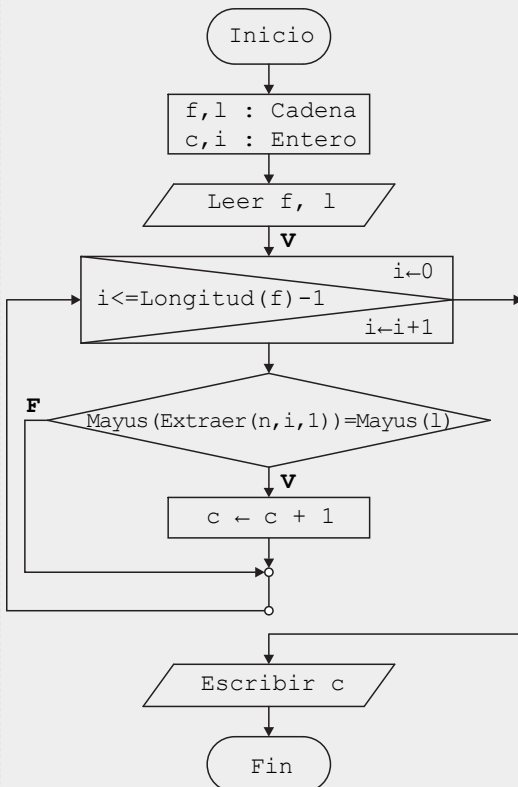


Diagrama de flujo



Algoritmo

Inicio

//Variables

f, l : Cadena
c, i : Entero

//Entrada

Leer f, l

//Proceso

Para i ← 0 Hasta Longitud(f) - 1 Inc 1

Si Mayus(Extraer(n, i, 1)) = Mayus(l)

Entonces

c ← c + 1

Fin Si

Fin Para

//Salida

Escribir c

Fin

Codificación:

```

<?php

//Variables
$f = ""; $l = "";
$c = 0; $i = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $f = $_POST["txtf"];
    $l = $_POST["txtl"];

    //Proceso
    for($i = 0;$i<=strlen($f)-1;$i++){
        if(substr(strtoupper($f),$i,1) == strtoupper($l))
            $c++;
    }
}
?>
<html>
<head>
<title>Problema 93</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema93.php">
    <table width="344" border="0">

        <tr>
            <td colspan="2"><strong>Problema 93</strong> </td>
        </tr>
        <tr>
            <td width="86">Frase</td>
            <td width="172"><textarea name="txtf" cols="40" rows="5"
id="txtf"><?=$f?></textarea></td>
        </tr>
        <tr>
            <td>Letra</td>
            <td><input name="txtl" type="text" id="txtl" value="<?=$l?>"
/>
            </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Cantidad</td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>" /></td>

```

```
</tr>

<tr>
  <td>&nbsp;</td>
  <td>
    <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
  </td>
</tr>
</table>
</form>
</body>
</html>
```

Problema n.º 94

Enunciado: Dada una frase, devolver la frase sin espacio en blancos.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese una frase; luego, que el sistema devuelva la frase sin espacios en blancos.

Entrada

- Frase (f1)

Salida

- Frase sin espacios en blanco (f2)

Diseño:**Interfaz de usuario**

Problema 94 - Windows Internet Explorer

http://localhost/Cap08/Problema94.php

Archivo Edición Ver Favoritos Herramientas Ayuda

Problema 94

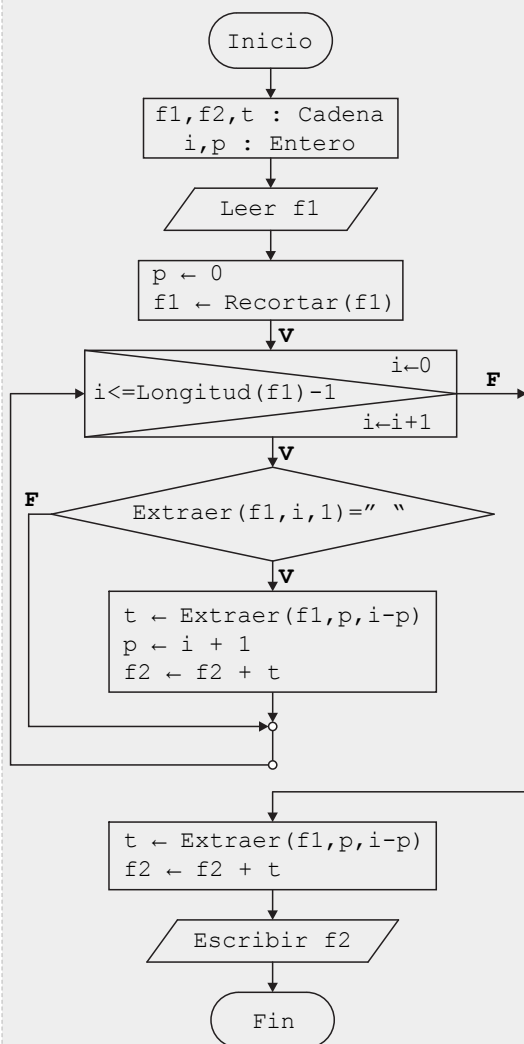
Problema 94

Frase: Paso a paso se llega a lejos

Frase sin espacios: Pascapasosellegaalejos

Calcular

Diagrama de flujo



Algoritmo

Pseudocódigo

Inicio

//Variables

f1, f2, t : Cadena
i, p : Entero

//Entrada

Leer f1

//Proceso

p ← 0
f1 ← Recortar(f1)

Para i←0 Hasta Longitud(f)-1 Inc 1

Si Extraer(f1, i, 1) = " " Entonces

t ← Extraer(f1, p, i-p)
p ← i + 1
f2 ← f2 + t

Fin Si

Fin Para

t ← Extraer(f1, p, i-p)
f2 ← f2 + t

//Salida

Escribir f2

Fin

Codificación:

```

<?php
//Variables
$f1 = ""; $f2 = ""; $t = "";
$i = 0; $p = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $f1 = $_POST["txtf1"];

    //Proceso
    $p = 0;
    $f1 = trim($f1);
    for($i = 0; $i<strlen($f1); $i++){
        if(substr($f1,$i, 1) == " ") {
            $t = substr($f1, $p, $i - $p);
            $p = $i + 1;
            $f2 = $f2 . $t;
        }
    }

    $t = substr($f1, $p, $i - $p);
    $f2 = $f2 . $t;
}
?>
<html>
<head>
<title>Problema 94</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema94.php">
    <table width="344" border="0">

        <tr>
            <td colspan="2"><strong>Problema 94</strong> </td>
        </tr>
        <tr>
            <td width="86">Frase</td>
            <td width="172"><textarea name="txtf1" cols="40" rows="5"
id="txtf1"><?=$f1?></textarea></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Frase sin espacios </td>

```

```

        <td><textarea name="txtf2" cols="40" rows="5" class="TextoFondo"
id="txtf2"><?=$f2?>
        </textarea></td>
    </tr>

    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 95

Enunciado: Dada una frase, devuelva la frase en forma encriptada, usando el método de convertir a los siguientes caracteres del ASCII; ejemplo, si el carácter es A = 65, devolverá B=66.

Análisis: Para la solución de este problema, se requiere que el usuario ingrese una frase; luego, que el sistema devolverá la frase en formato encriptado.

Entrada

- Frase (f1)

Salida

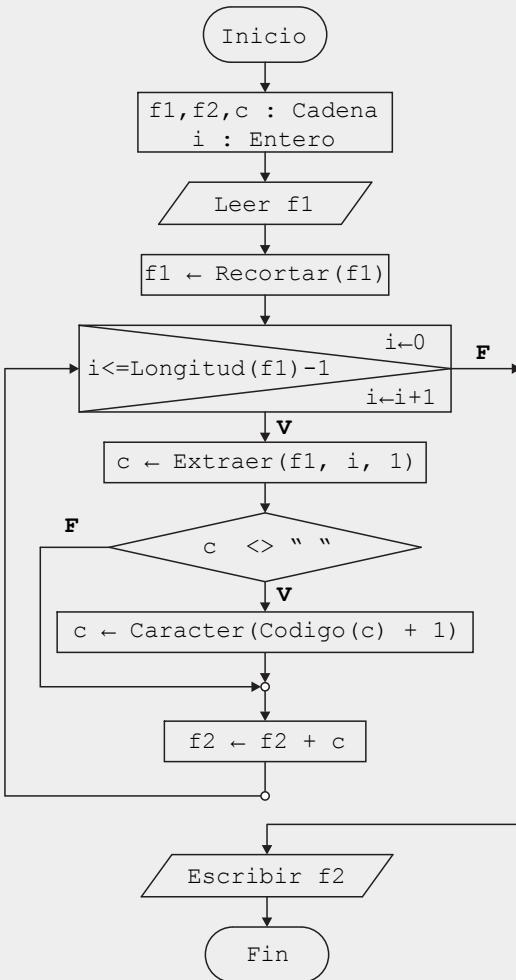
- Frase encriptada (f2)

Diseño:

Interfaz de usuario

Algoritmo

Diagrama de flujo



Pseudocódigo

Inicio

//Variables

f1, f2, c : Cadena
i : Entero

//Entrada

Leer f1

//Proceso

f1 ← Recortar(f1)

Para i←0 Hasta Longitud(f1)-1 Inc 1

c ← Extraer(f1, i, 1)

Si c <> " " Entonces

c ← Caracter(Codigo(c) + 1)

Fin Si

f2 ← f2 + c

Fin Para

//Salida

Escribir f2

Fin

Codificación:

```

<?php

//Variables
$f1 = ""; $f2 = ""; $c = "";
$i = 0; $p = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $f1 = $_POST["txtf1"];

    //Proceso
    $p = 0;
    $f1 = trim($f1);
}

```

```

        for($i = 0; $i<strlen($f1); $i++){
            $c = substr($f1,$i,1);
            if($c != " ") {
                $c = chr(ord($c) + 1);
            }
            $f2 .= $c;
        }
    }
    ?>
<html>
<head>
<title>Problema 95</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema95.php">
    <table width="344" border="0">

        <tr>
            <td colspan="2"><strong>Problema 95</strong> </td>
        </tr>
        <tr>
            <td width="86">Frase</td>
            <td width="172"><textarea name="txtf1" cols="40" rows="5"
id="txtf1"><?=$f1?></textarea></td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Frase encriptada </td>
            <td><textarea name="txtf2" cols="40" rows="5" class="TextoFondo"
id="txtf2"><?=$f2?></textarea></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

8.12 Problemas propuestos

Los siguientes problemas le servirán para medir su aprendizaje, es importante que los resuelva.

Propuesto n.º 61

Enunciado: Dado el nombre de una persona obtenga el mensaje «Bienvenido, Sr(a) Gustavo, a su tienda de preferencia».

Propuesto n.º 62

Enunciado: Dado un nombre, obtenga el nombre en forma invertida, así: «Julio» invertido es «oiluJ».

Propuesto n.º 63

Enunciado: Dada una frase, devuelva la frase con asteriscos en lugar de espacios en blancos.

Propuesto n.º 64

Enunciado: Dada una letra, determine si está en minúscula o mayúscula.

Propuesto n.º 65

Enunciado: Lea una frase y una palabra, y determine si existe o no la palabra en la frase.

Propuesto n.º 66

Enunciado: Dada una palabra, determinar si es palíndromo (una palabra es palíndromo si se lee igual de izquierda a derecha o de derecha a izquierda), por ejemplo ANA.

Propuesto n.º 67

Enunciado: Dada una frase, determine cuántas palabras palíndromos ha ingresado.

Propuesto n.º 68

Enunciado: Dada una frase, determine cuántas palabras se repiten.

Propuesto n.º 69

Enunciado: Cree el algoritmo para encriptar una frase con el valor del carácter ASCII, sumando 2 posiciones.

Propuesto n.º 70

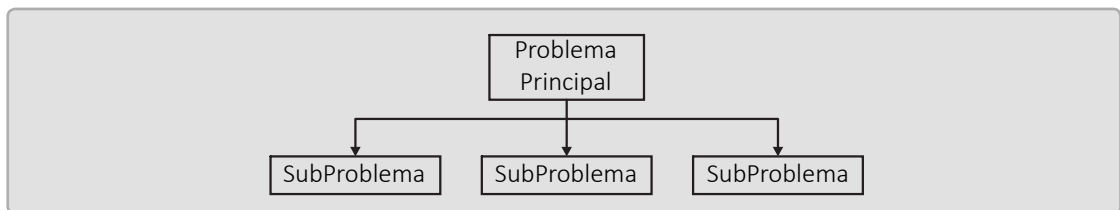
Enunciado: Cree el algoritmo para desencriptar la frase generada por el algoritmo anterior.

SubAlgoritmos (procedimientos y funciones)

9.1 Introducción

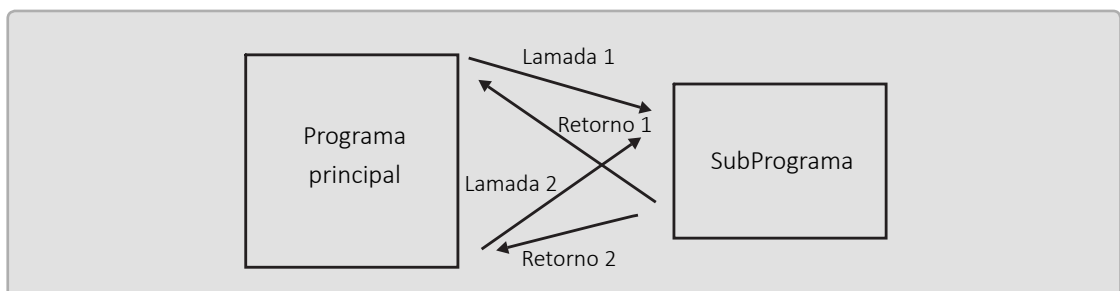
Una frase bastante usada en el mundo de la informática, para resolver problemas complejos, es: «Divide y vencerás». Esta es aplicada al tema de subalgoritmos (subprogramas), y consiste en dividir un problema grande en problemas más pequeños que se encargarán de resolver temas específicos. Los subalgoritmos (subprogramas) se dividen en dos tipos de procedimientos (subrutinas) y funciones, que evitarán la duplicidad de código y ayudarán a crear módulos más pequeños para un mejor mantenimiento, y permite reutilizarlo muchas veces.

El método para diseñar la solución de un problema principal (*main*) en subproblemas se conoce como diseño descendente (*top-down design*), difundida por la programación modular.



El problema principal corresponde al programa o algoritmo principal, y la solución de los subproblemas mediante subprogramas (procedimientos y funciones), en el lenguaje algorítmico se conoce como subalgoritmos.

El subprograma recibe datos y es invocado desde el programa principal, después de terminar el proceso que tuvo que realizar el subprograma devuelve el resultado correspondiente al programa principal.



9.2 Procedimientos

Los procedimientos se caracterizan por realizar una tarea específica y no retornar un resultado; sin embargo, sí es posible implementar que devuelva resultados por intermedio de parámetros llamados de salida o por referencia.

Pseudocódigo

```
//Crear un procedimiento
Procedimiento Proc1 (E:Param1:Entero)

    <Instrucciones>

Fin Procedimiento

//Invocar el procedimiento
Llamar Proc1(10)
```

PHP

```
//Función que no retorna ningun valor (void)
function Proc1($Param1) {

    <Instrucciones>;

}

`Invocar la función
Proc1(10);
```

9.3 Funciones

Son más conocidos por devolver un valor como resultado de la tarea realizada; los lenguajes de programación incorporan funciones que realizan algunas tareas ya programadas, conocidas como funciones internas, pero las funciones programadas por el usuario (programador) se conocen como externas o funciones definidas por el usuario (FDU).

Pseudocódigo

```
//Crear una función
Funcion Func1 (E:Param1:Entero):Cadena

    <Instrucciones>

    Retorna <Valor>

Fin Funcion

//Invocar la función
c ← Func1(10)
```

PHP

```
//Crear una método que retorna un valor
function Func1($Param1) {

    <Instrucciones>;

    return <Valor>;

}

//Invocar el método
$c = Func1(10);
```

9.4 Paso de parámetros

Muchas veces los procedimientos y funciones requieren que le envíen una lista de valores llamados parámetros (argumentos), para usarlos en la solución de la tarea encomendada. Los parámetros son variables, muchas veces de entrada (reciben valores) y de salida (devuelven resultados) o ambos de entrada/salida.

Estos parámetros también toman el nombre de **parámetros por valor** (entrada) y **parámetros por referencias** (salida).

9.5 Parámetros por valor (entrada)

Los valores que se envían a los parámetros son asignados como una copia de los valores originales, desconectando el programa principal con el subprograma; es decir, si los valores de los parámetros cambian dentro del subprograma no afecta al programa principal.

Pseudocódigo

```
//Crear una función
Funcion Incrementar(E:N:Entero):Entero

    N ← N + 1 //Modifica el valor de N

    Retorna N

Fin Funcion

//Invocar la función
Num ← 5
Res ← Incrementar(Num) //El valor de Num se copia en N
Imprimir Num //su valor sigue siendo 5
Imprimir Res //su valor es 6
```

PHP

```
//Crear una función
function Incrementar($N) {

    $N = $N + 1; //Modifica el valor de N

    return $N;

}

//Invocar el método
$Num = 5;
$Res = Incrementar($Num); //El valor de Num se copia en N
echo($Num); //su valor sigue siendo 5
echo($Res); //su valor es 6
```

9.6 Parámetros por referencia (salida)

Se asignan las referencias de las variables (dirección de memoria de la variable) a los parámetros, conectando el programa principal con el subprograma; es decir, si los valores de los parámetros cambian dentro del subprograma, afecta a las variables del programa principal.

Pseudocódigo

```
//Crear una función
Funcion Incrementar(S:N:Entero):Entero

    N ← N + 1 //Modifica el valor de N

    Retorna N

Fin Funcion

//Invocar la función
Num ← 5
Res ← Incrementar(Num) //El parámetro N hace referencia a Num
Imprimir Num //su valor ahora es 6
Imprimir Res //su valor es 6
```

PHP

```
//Crear una función
function Incrementar(&$N) {

    $N = $N + 1; //Modifica el valor de N

    return $N;

}

//Invocar la función
$Num = 5;
$Res = Incrementar($Num); //El parámetro N hace referencia a Num
echo($Num); //su valor ahora es 6
echo($Res); //su valor es 6
```

Problema n.º 96

Enunciado: Dados dos números enteros, hallar la suma. Cree una función para resolver el problema.

Sumar(E:Num1:Entero, E:Num2:Entero):Entero

Interfaz de usuario

Problema 96 - Windows Internet Explorer

http://localhost/Cap09/Problema96.php

Archivo Edición Ver Favoritos Herramientas Ayuda

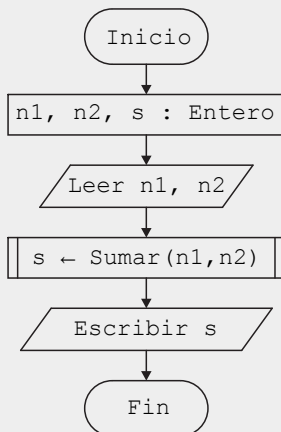
Problema 96

Problema 96

Numero 1

Numero 2

Suma

Algoritmo**Diagrama de flujo****Principal****Pseudocódigo****Principal****Inicio**

//Variables

`n1, n2, s : Entero`

//Entrada

Leer n1, n2

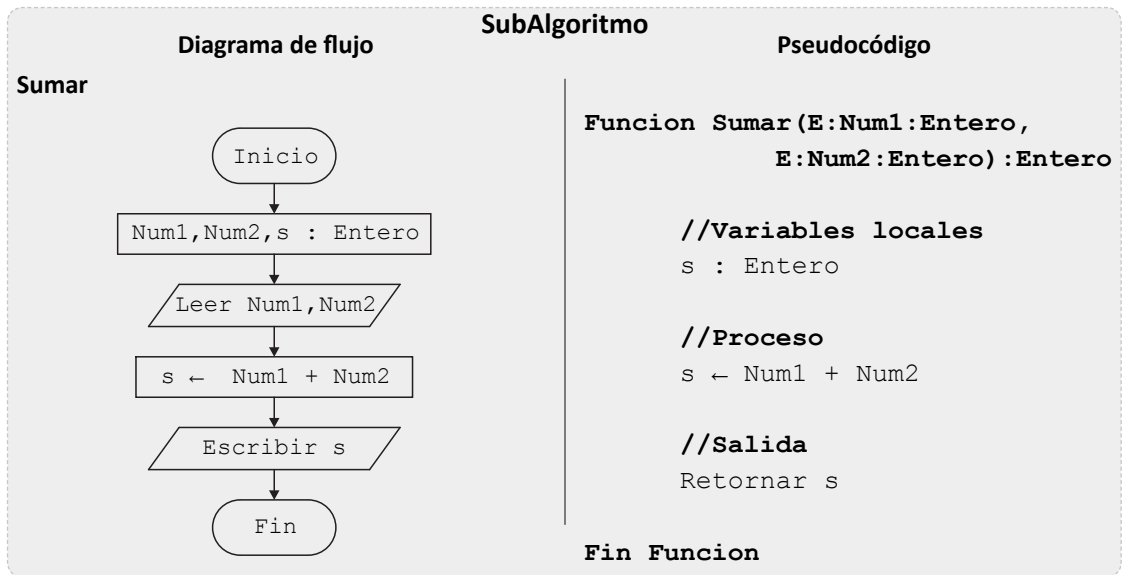
//Proceso

`s ← Sumar(n1 + n2)`

//Salida

Escribir s

Fin



Codificación:

```

<?php

//Principal

//Variables
$n1 = 0; $n2 = 0; $s = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n1 = $_POST["txtn1"];
    $n2 = $_POST["txtn2"];

    //Proceso
    $s = Sumar($n1, $n2);
}

//Función Sumar
function Sumar($Num1, $Num2) {
    //Variables
    $s;

    //Proceso
    $s = $Num1 + $Num2;

    //Salida
    return $s;
}

?>
<html>

```

```

<head>
<title>Problema 96</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema96.php">
    <table width="242" border="0">

        <tr>
            <td colspan="2"><strong>Problema 96 </strong></td>
        </tr>
        <tr>
            <td width="79">Numero 1 </td>
            <td width="144"><input name="txtn1" type="text" id="txtn1"
value="<?=$n1?>"></td>
        </tr>
        <tr>
            <td>Numero 2 </td>
            <td><input name="txtn2" type="text" id="txtn2" value="<?=$n2?>"></
td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Suma</td>
            <td><input name="txts" type="text" class="TextoFondo" id="txts"
value="<?=$s?>"></td>
        </tr>

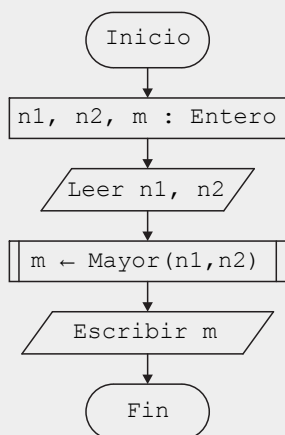
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 97

Enunciado: Dados dos números enteros diferentes, devolver el número mayor. Cree una función para resolver el problema.

Mayor(E:n1:Entero, E:n2:Entero):Entero

Interfaz de usuario
Diagrama de flujo**Principal****Algoritmo****Pseudocódigo****Principal****Inicio****//Variables**

`n1, n2, m : Entero`

//Entrada

Leer `n1, n2`

//Proceso

`m ← Mayor(n1,n2)`

//Salida

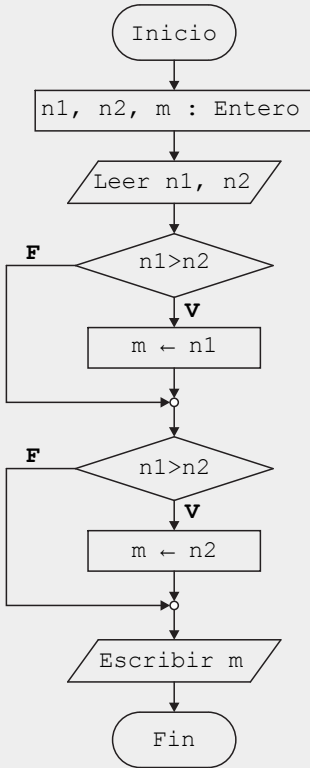
Escribir `m`

Fin

SubAlgoritmo

Diagrama de flujo

Mayor



Pseudocódigo

**Funcion Mayor(E:n1:Entero,
E:n2:Entero):Entero**

//Variables locales

m : Entero

//Proceso

Si n1 > n2 Entonces

m ← n1

Fin Si

Si n2 > n1 Entonces

m ← n2

Fin Si

//Salida

Retorna m

Fin Funcion

Codificación:

```

<?php

//Principal

//Variables
$n1 = 0; $n2 = 0; $m = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n1 = $_POST["txtn1"];
    $n2 = $_POST["txtn2"];

    //Proceso
    $m = Mayor($n1, $n2);
}

//Método Mayor
function Mayor($n1, $n2){
    //Variables
    $m = 0;
  
```

```

//Proceso
if($n1 > $n2)
    $m = $n1;

if($n2 > $n1)
    $m = $n2;

//Salida
return $m;
}
?>
<html>
<head>
<title>Problema 97</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema97.php">
    <table width="242" border="0">

        <tr>
            <td colspan="2"><strong>Problema 97</strong> </td>
        </tr>
        <tr>
            <td width="79">Numero 1 </td>
            <td width="144"><input name="txtn1" type="text" id="txtn1"
value="<?=$n1?>"></td>
        </tr>
        <tr>
            <td>Numero 2 </td>
            <td><input name="txtn2" type="text" id="txtn2" value="<?=$n2?>"></
td>
        </tr>
        <tr>
            <td>Mayor</td>
            <td><input name="txtm" type="text" class="TextoFondo" id="txtm"
value="<?=$m?>"></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

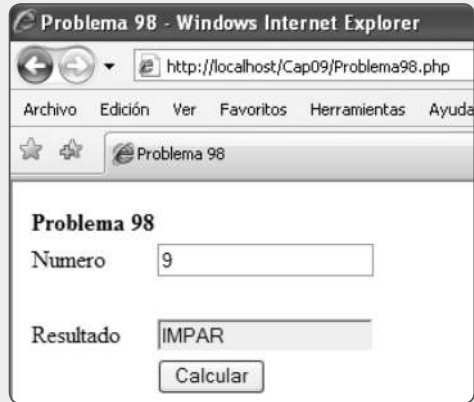
```

Problema n.º 98

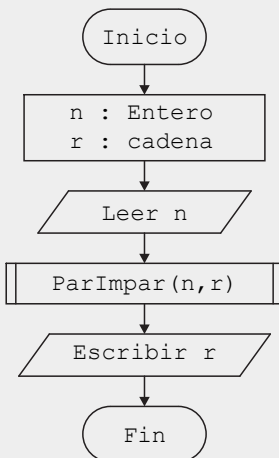
Enunciado: Determinar si un número entero es par o impar. Cree un procedimiento para resolver el problema.

ParImpar(E:num:Entero, S:res:Entero)

Interfaz de usuario

**Algoritmo****Diagrama de flujo**

Principal

**Pseudocódigo**

Principal

Inicio

//Variables

n : Entero
r : Cadena

//Entrada

Leer n

//Proceso

ParImpar(n, r)

//Salida

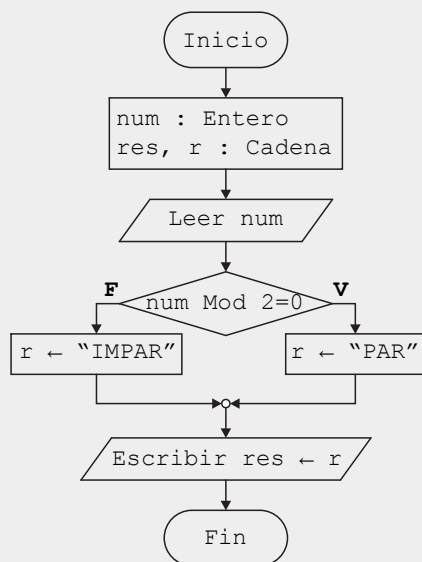
Escribir r

Fin

SubAlgoritmo

Diagrama de flujo

ParImpar



Pseudocódigo

Procedimiento ParImpar(E:num:Entero,
S:res:Cadena)

//Variables locales

r : Cadena

//Proceso

Si num Mod 2 = 0 Entonces

 r ← "PAR"

SiNo

 r ← "IMPAR"

Fin Si

//Salida

res ← r

Fin Procedimiento

Codificación:

```

<?php

//Principal

//Variables
$n = 0;
$r = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = $_POST["txttn"];

    //Proceso
    ParImpar($n,$r);
}

//Función ParImpar
function ParImpar($num, &$res) {
    //Variables
    $r = "";

    //Proceso
    if($num % 2 == 0){
        $r = "PAR";
    }else{
        $r = "IMPAR";
    }
}
  
```

```

        //Salida
        $res = $r;
    }

?>
<html>
<head>
<title>Problema 98</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema98.php">
    <table width="242" border="0">

        <tr>
            <td colspan="2"><strong>Problema 98</strong> </td>
        </tr>
        <tr>
            <td width="79">Numero</td>
            <td width="144"><input name="txtn" type="text" id="txtn"
value="<?=$n?>"></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Resultado</td>
            <td><input name="txtr" type="text" class="TextoFondo" id="txtr"
value="<?=$r?>"></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 99

Enunciado: Dado un número, determinar cuantos dígitos tiene. Cree una función para resolver el problema.

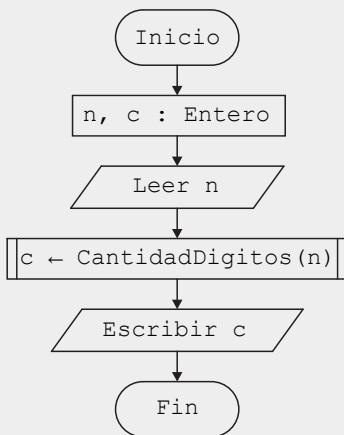
CantidadDigitos(E:num:Entero):Entero

Interfaz de usuario

Algoritmo

Diagrama de flujo

Principal



Pseudocódigo

Principal

Inicio

//Variables

n, c : Entero

//Entrada

Leer n

//Proceso

c ← CantidadDigitos(n)

//Salida

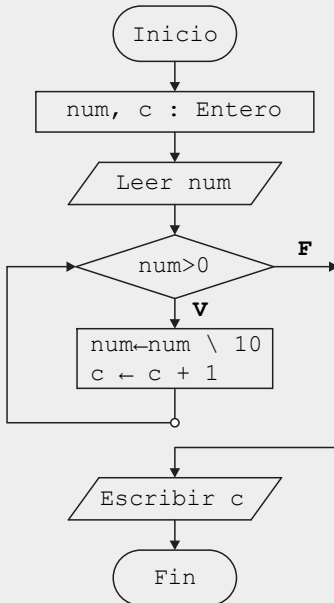
Escribir c

Fin

SubAlgoritmo

Diagrama de flujo

CantidadDigitos



Pseudocódigo

Funcion CantidadDigitos
(E:num:Entero):Entero

//Variables locales
 c : Entero

//Proceso
 Mientras num>0
 num ← num \ 10
 c ← c + 1
 Fin Mientras

//Salida
 Retornar c

Fin Funcion

Codificación:

```

<?php

//Principal

//Variables
$n = 0; $c = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = $_POST["txtn"];

    //Proceso
    $c = CantidadDigitos($n);
}

//Función CantidadDigitos
function CantidadDigitos($num) {
    //Variables
    $c=0;

    //Proceso
    while($num > 0) {
        $num = (int)($num / 10);
        $c += 1;
    }
}
  
```

```

        //Salida
        return $c;
    }
    ?>
<html>
<head>
<title>Problema 99</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema99.php">
    <table width="242" border="0">

        <tr>
            <td colspan="2"><strong>Problema 99</strong> </td>
        </tr>
        <tr>
            <td width="79">Numero</td>
            <td width="144"><input name="txtn" type="text" id="txtn"
value="<?=$n?>"></td>
        </tr>

        <tr>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
        </tr>
        <tr>
            <td>Cant. Digitos </td>
            <td><input name="txtc" type="text" class="TextoFondo" id="txtc"
value="<?=$c?>"></td>
        </tr>

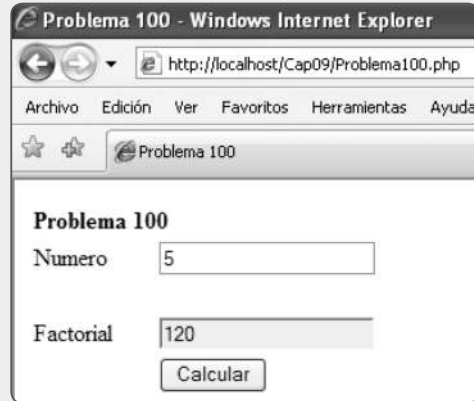
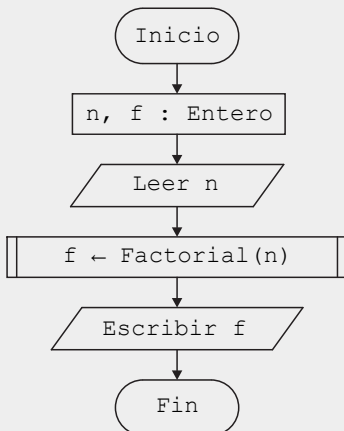
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 100

Enunciado: Crear un algoritmo para hallar el factorial de un número, el factorial es el producto de todos los números consecutivos, desde la unidad hasta el número. Por ejemplo, factorial de 3! (se denota !) es $1 \times 2 \times 3 = 6$. Cree una función para resolver el problema.

Factorial(E:num:Entero):Entero

Interfaz de usuario**Algoritmo****Diagrama de flujo****Principal****Pseudocódigo****Principal****Inicio**

//Variables

n, f : Entero

//Entrada

Leer n

//Proceso

f ← Factorial(n)

//Salida

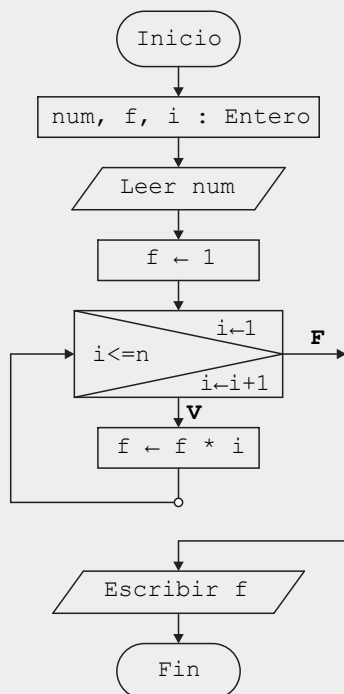
Escribir f

Fin

SubAlgoritmo

Diagrama de flujo

Factorial



Pseudocódigo

Funcion Factorial (E:num:Entero) :Entero

//Variables locales

f, i : Entero

//Proceso

f ← 1

Para i←1 Hasta num Inc 1

f ← f * i

Fin Para

//Salida

Retornar f

Fin Funcion

Codificación:

```

<?php
//Principal

//Variables
$n = 0; $f = 0; $i = 0;

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n = $_POST["txtn"];

    //Proceso
    $f = Factorial($n);
}

//Función Factorial
function Factorial($num) {
    //Variables
    $i = 0; $f = 0;

    //Proceso
    $f = 1;
    for($i = 1; $i<=$num; $i++)
        $f *= $i;

    //Salida
    return $f;
}
?>
<html>
<head>
<title>Problema 100</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema100.php">
    <table width="242" border="0">

        <tr>
            <td colspan="2"><strong>Problema 100</strong> </td>
        </tr>
        <tr>
            <td width="79">Numero</td>
            <td width="144"><input name="txtn" type="text" id="txtn"
value="<?=$n?>"></td>
        </tr>

        <tr>

```

```

        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>Factorial </td>
        <td><input name="txtf" type="text" class="TextoFondo" id="txtf"
value="<?=$f?>"></td>
    </tr>

    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

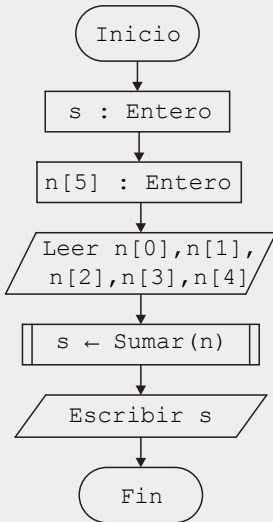
Problema n.º 101

Enunciado: Dados 5 números, obtener la suma. Cree una función para resolver el problema.

Sumar(E:num[]):Entero:Entero

Interfaz de usuario

The screenshot shows a web browser window titled "Problema 101 - Windows Internet Explorer". The address bar shows the URL "http://localhost/Cap09/Problema101.php". The browser's menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The page content is titled "Problema 101" and features five input fields labeled "Numero 1" through "Numero 5", each containing the number "2". Below these fields is a "Suma" field containing the number "10" and a "Calcular" button.

Algoritmo**Diagrama de flujo****Principal****Pseudocódigo****Principal****Inicio****//Variables**

s : Entero

//Arreglos (Vector)

n[5] : Entero

//Entrada

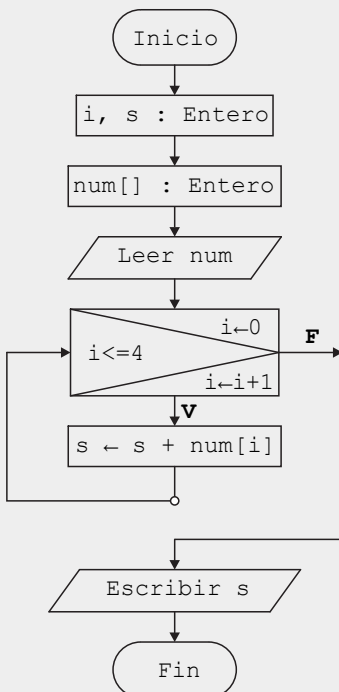
Leer n[0],n[1],n[2],n[3],n[4]

//Proceso

s ← Sumar(n)

//Salida

Escribir s

Fin**SubAlgoritmo****Diagrama de flujo****Sumar****Pseudocódigo****Funcion Sumar(E:num[]:Entero):Entero****//Variables locales**

i, s : Entero

//Proceso

Para i ← 0 Hasta 4 Inc 1

s ← s + num[i]

Fin Para

//Salida

Retornar s

Fin Funcion

Codificación:

```
<?php

//Principal

//Variables
$s = 0; $i = 0;

//Arreglo
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0] = $_POST["txtn1"];
    $n[1] = $_POST["txtn2"];
    $n[2] = $_POST["txtn3"];
    $n[3] = $_POST["txtn4"];
    $n[4] = $_POST["txtn5"];

    //Proceso
    $s = Sumar($n);
}

//Función Sumar
function Sumar($num) {
    //Variables
    $s = 0; $i = 0;

    //Proceso
    for($i = 0; $i <= 4; $i++)
        $s += $num[$i];

    //Salida
    return $s;
}
?>
<html>
<head>
<title>Problema 101</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema101.php">
    <table width="242" border="0">
```

```

    <tr>
        <td colspan="2"><strong>Problema 101</strong> </td>
    </tr>
    <tr>
        <td width="79">Numero 1 </td>
        <td width="144"><input name="txtn1" type="text" id="txtn1"
value="<?=$n[0]?>"></td>
    </tr>
    <tr>
        <td>Numero 2 </td>
        <td><input name="txtn2" type="text" id="txtn2" value="<?=$n[1]?>"></
td>
    </tr>
    <tr>
        <td>Numero 3 </td>
        <td><input name="txtn3" type="text" id="txtn3" value="<?=$n[2]?>"></
td>
    </tr>
    <tr>
        <td>Numero 4 </td>
        <td><input name="txtn4" type="text" id="txtn4" value="<?=$n[3]?>"></
td>
    </tr>
    <tr>
        <td>Numero 5 </td>
        <td><input name="txtn5" type="text" id="txtn5" value="<?=$n[4]?>"></
td>
    </tr>

    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>Suma </td>
        <td><input name="txts" type="text" class="TextoFondo" id="txts"
value="<?=$s?>"></td>
    </tr>

    <tr>
        <td>&nbsp;</td>
        <td>
            <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

Problema n.º 102

Enunciado: Ordene 4 números usando el método de ordenación por intercambio (burbuja). Cree un procedimiento para resolver el problema.

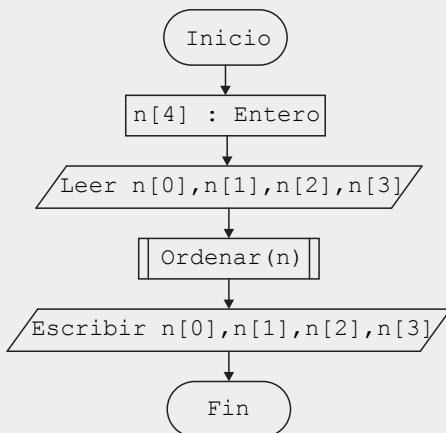
Ordenar (S:num[:Entero])

Interfaz de usuario

The screenshot shows a web browser window titled 'Problema 102 - Windows Internet Explorer'. The address bar shows 'http://localhost/Cap09/Problema102.php'. The browser menu includes 'Archivo', 'Edición', 'Ver', 'Favoritos', 'Herramientas', and 'Ayuda'. The page content is titled 'Problema 102' and contains five input fields labeled 'Numero 1' through 'Numero 5' with values 1, 2, 3, 4, and 5 respectively. A 'Calcular' button is located at the bottom right of the input fields.

Algoritmo**Diagrama de flujo**

Principal

**Pseudocódigo**

Principal

Inicio

//Arreglos (Vector)

n[4] : Entero

//Entrada

Leer n[0],n[1],n[2],n[3]

//Proceso

Ordenar(n)

//Salida

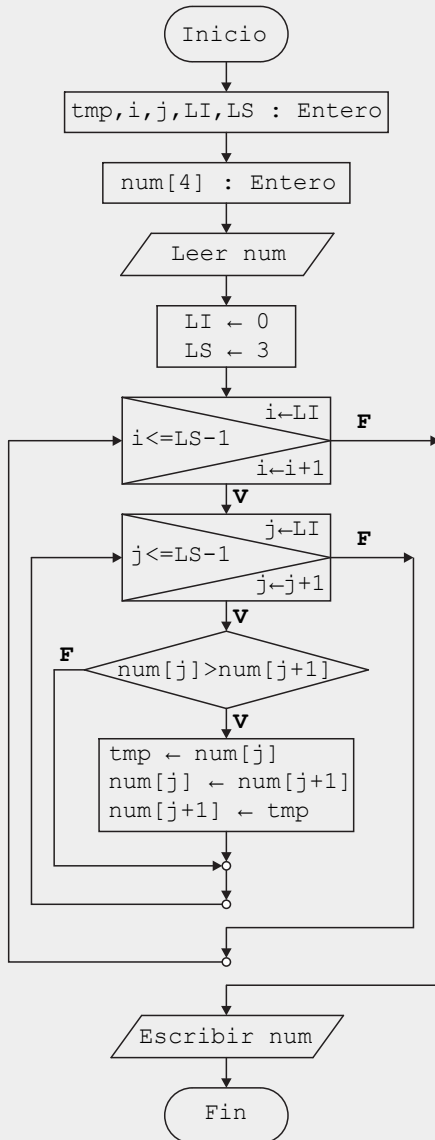
Escribir n[0],n[1],n[2],n[3]

Fin

SubAlgoritmo

Diagrama de flujo

Ordenar



Pseudocódigo

Procedimiento Ordenar(S:num[]:Entero)**//Variables locales**

tmp,i,j,LI,LS : Entero

//Proceso

LI ← 0

LS ← 3

Para i←LI Hasta LS-1 Inc 1

Para j←LI Hasta LS-1 Inc 1

Si num[j]>num[j+1] Entonces

tmp ← num[j]

num[j] ← num[j+1]

num[j+1] ← tmp

Fin Si

Fin Para

Fin Para

//Salida

Escribir num

Fin Procedimiento

Codificación:

```
<?php

//Principal

//Arreglo
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0] = $_POST["txtn1"];
    $n[1] = $_POST["txtn2"];
    $n[2] = $_POST["txtn3"];
    $n[3] = $_POST["txtn4"];
    $n[4] = $_POST["txtn5"];

    //Proceso
    Ordenar($n);
}

//Funcion Ordenar
function Ordenar (&$num) {
    //Variables
    $tmp = 0; $i = 0; $j = 0; $LI = 0; $LS = 0;

    //Arreglos
    $n = $num;

    //Proceso
    $LI = 0;
    $LS = count($n) -1;

    for($i = $LI; $i <= $LS - 1; $i++){
        for($j = $LI; $j <= $LS - 1; $j++){
            if($n[$j] > $n[$j + 1]){
                $tmp = $n[$j];
                $n[$j] = $n[$j + 1];
                $n[$j + 1] = $tmp;
            }
        }
    }

    //Salida
    $num = $n;
}

?>
<html>
<head>
<title>Problema 102</title>
<style type="text/css">
```

```

<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>
</head>
<body>
<form method="post" action="Problema102.php">
    <table width="242" border="0">

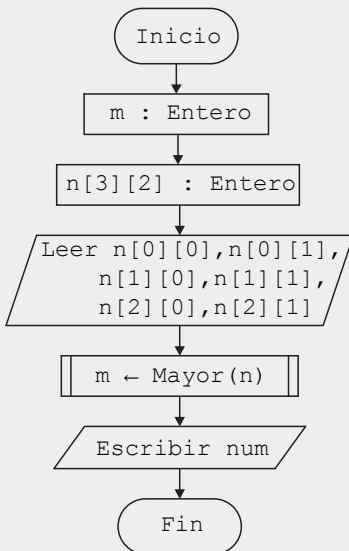
        <tr>
            <td colspan="2"><strong>Problema 102 </strong></td>
        </tr>
        <tr>
            <td width="79">Numero 1 </td>
            <td width="144"><input name="txtn1" type="text" id="txtn1"
value="<?=$n[0]?>"></td>
        </tr>
        <tr>
            <td>Numero 2 </td>
            <td><input name="txtn2" type="text" id="txtn2" value="<?=$n[1]?>"></
td>
        </tr>
        <tr>
            <td>Numero 3 </td>
            <td><input name="txtn3" type="text" id="txtn3" value="<?=$n[2]?>"></
td>
        </tr>
        <tr>
            <td>Numero 4 </td>
            <td><input name="txtn4" type="text" id="txtn4" value="<?=$n[3]?>"></
td>
        </tr>
        <tr>
            <td>Numero 5 </td>
            <td><input name="txtn5" type="text" id="txtn5" value="<?=$n[4]?>"></
td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td>
                <input name="btnCalcular" type="submit" id="btnCalcular"
value="Calcular" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

Problema n.º 103

Enunciado: Ingrese 6 números en una matriz de 3x2 y obtenga el número mayor ingresado. Cree una función para resolver el problema.

Mayor(E:num[]:Entero):Entero

Interfaz de usuario
Algoritmo**Diagrama de flujo****Principal****Pseudocódigo****Principal****Inicio**

//Variables

m : Entero

//Arreglos (Matriz)

n[3][2] : Entero

//Entrada

Leer n[0][0], n[0][1],
n[1][0], n[1][1],
n[2][0], n[2][1]

//Proceso

m ← Mayor(n)

//Salida

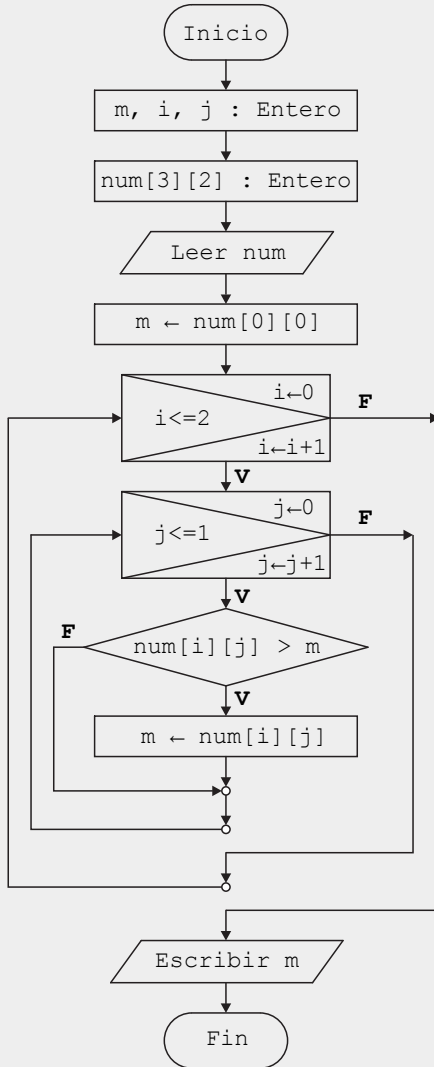
Escribir m

Fin

SubAlgoritmo

Diagrama de flujo

Mayor



Pseudocódigo

Funcion Mayor (E:num[] []:Entero) :Entero**//Variables locales****m, i, j : Entero****//Arreglos (Matriz)****num[3][2] : Entero****//Proceso****m ← num[0][0]****Para i←0 Hasta 2 Inc 1****Para j←0 Hasta 1 Inc 1****Si num[i][j]>m Entonces****m ← num[i][j]****Fin Si****Fin Para****Fin Para****//Salida****Escribir m****Fin Funcion**

Codificación:

```

<?php

//Principal

//Variables
$m = 0;

//Arreglo
$n = array();

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $n[0][0] = $_POST["txtn00"];
    $n[0][1] = $_POST["txtn01"];
    $n[1][0] = $_POST["txtn10"];
    $n[1][1] = $_POST["txtn11"];
    $n[2][0] = $_POST["txtn20"];
    $n[2][1] = $_POST["txtn21"];

    //Proceso
    $m = Mayor($n);
}

//Función Mayor
function Mayor($num) {
    //Variables
    $m = 0; $i = 0; $j = 0;

    //Proceso
    $m = $num[0][0];
    for($i = 0;$i<=2;$i++){
        for($j = 0; $j<=1;$j++){
            if($num[$i][$j] > $m)
                $m = $num[$i][$j];
        }
    }

    //Salida
    return $m;
}
?>
<html>
<head>
<title>Problema 103</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}

```

```

-->
</style>
</head>
<body>
<form method="post" action="Problema103.php">
  <table width="211" border="0">

    <tr>
      <td colspan="4"><strong>Problema 103</strong></td>
    </tr>
    <tr>
      <td colspan="4"><strong>Arreglo 3X2</strong> </td>
    </tr>
    <tr>
      <td width="36"><input name="txtn00" type="text" id="txtn00"
value="<?=$n[0][0]?" size="5"></td>
      <td width="46"><input name="txtn01" type="text" id="txtn01"
value="<?=$n[0][1]?" size="5"></td>
      <td width="41">Mayor</td>
      <td width="47"><input name="txtm" type="text" class="TextoFondo"
id="txtm" value="<?=$m?" size="5"></td>
    </tr>
    <tr>
      <td><input name="txtn10" type="text" id="txtn10" value="<?=$n[1]
[0]?" size="5"></td>
      <td><input name="txtn11" type="text" id="txtn11" value="<?=$n[1]
[1]?" size="5"></td>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>
    <tr>
      <td><input name="txtn20" type="text" id="txtn20" value="<?=$n[2]
[0]?" size="5"></td>
      <td><input name="txtn21" type="text" id="txtn21" value="<?=$n[2]
[1]?" size="5"></td>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>
    <tr>
      <td colspan="2"><input name="btnCalcular" type="submit"
id="btnCalcular" value="Calcular" /></td>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>
  </table>
</form>
</body>
</html>

```

Problema n.º 104

Enunciado: Dada una frase, devolver la frase sin espacios en blancos. Cree una función para resolver el problema.

FraseSinEspacios(E:Frase:Cadena):Cadena

Interfaz de usuario

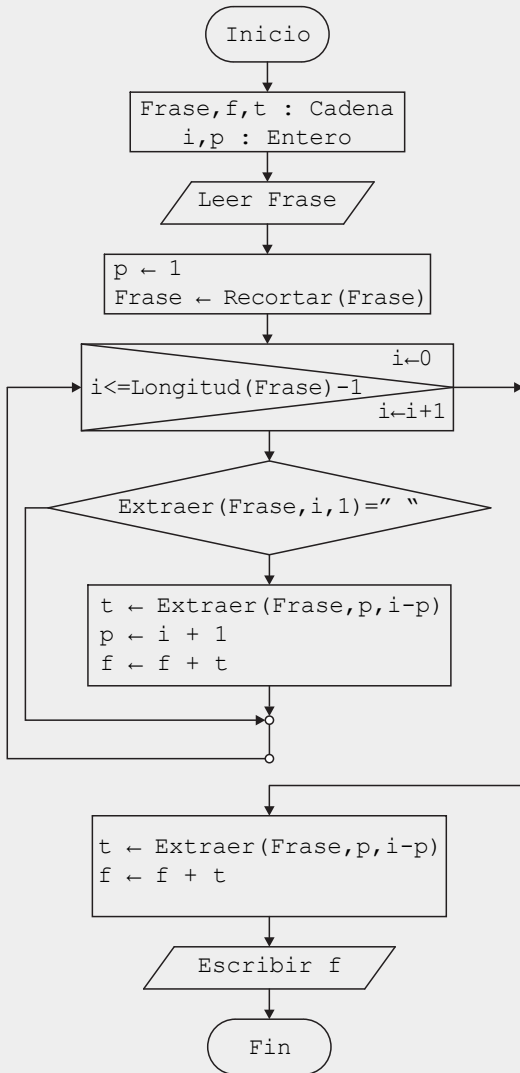


	Diagrama de flujo	Algoritmo	Pseudocódigo
Principal	<pre> graph TD Inicio([Inicio]) --> Declara[f1,f2 : Cadena] Declara --> Lee[/Leer f1/] Lee --> Procesa[f2←FraseSinEspacios(f1)] Procesa --> Escribe[/Escribir f2/] Escribe --> Fin([Fin]) </pre>	<pre> Principal Inicio //Variables f1,f2 : Cadena //Entrada Leer f1 //Proceso f2 ← FraseSinEspacios(f1) //Salida Escribir f2 Fin </pre>	

SubAlgoritmo

Diagrama de flujo

FrasesSinEspacios



Pseudocódigo

Funcion FraseSinEspacios
(E:Frased:Cadena) : Cadena

//Variables

f2, t : Cadena

i, p : Entero

//Proceso

p ← 1

Frased ← Recortar(Frased)

Para i ← 0 Hasta Longitud(Frased) - 1
 Inc 1

Si Extraer(Frased, i, 1) = " " Entonces

 t ← Extraer(Frased, p, i - p)

 p ← i + 1

 f ← f + t

Fin Si

Fin Para

 t ← Extraer(Frased, p, i - p)

 f ← f + t

//Salida

Retornar f

Fin Funcion

Codificación:

```

<?php

//Principal

//Variables
$f1 = ""; $f2 = "";

if(isset($_POST["btnCalcular"])) {
    //Entrada
    $f1 = $_POST["txtf1"];

    //Proceso
    $f2 = FraseSinEspacios($f1);
}

//Funcion FraseSinEspacios
function FraseSinEspacios($Frase) {
    //Variables
    $f = ""; $t = 0;
    $i = 0; $p = 0;

    //Proceso
    $p = 0;
    $Frase = trim($Frase);
    for($i = 0; $i<strlen($Frase); $i++){
        if(substr($Frase,$i, 1) == " ") {
            $t = substr($Frase, $p, $i - $p);
            $p = $i + 1;
            $f = $f . $t;
        }
    }

    $t = substr($Frase, $p, $i - $p);
    $f = $f . $t;

    //Salida
    return $f;
}
?>
<html>
<head>
<title>Problema 104</title>
<style type="text/css">
<!--
.TextoFondo {
    background-color: #CCFFFF;
}
-->
</style>

```

```
</head>
<body>
<form method="post" action="Problema104.php">
  <table width="334" border="0">
    <tr>
      <td colspan="2"><strong>Problema 104</strong> </td>
    </tr>
    <tr>
      <td>Frase</td>
      <td><textarea name="txtf1" cols="30" rows="5" id="txtf1"><?=$f1?></textarea></td>
    </tr>
    <tr>
      <td width="100">Frase sin espacio </td>
      <td width="216"><textarea name="txtf2" cols="30" rows="5" class="TextoFondo" id="txtf2"><?=$f2?></textarea></td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td><input name="btnCalcular" type="submit" id="btnCalcular" value="Calcular" /></td>
    </tr>
  </table>
</form>
</body>
</html>
```

9.7 Problemas propuestos

Los siguientes problemas le servirán para medir su aprendizaje, es importante que los resuelva.

Propuesto n.º 71

Enunciado: Hallar el Área y el Perímetro de un de un Cuadrado, cree un procedimiento para realizar dicha tarea.

Cuadrado (E:Lado:Real, S:Area:Real, S:Perimetro:Real)

Propuesto n.º 72

Enunciado: Dadas tres notas, obtenga el promedio de las dos notas mayores y cree un procedimiento para realizar dicha tarea.

Promedio(E:N1:Real,E:N2:Real,E:N3:Real,S:Promedio:Real)

Propuesto n.º 73

Enunciado: Dada la edad de una persona, determine en qué etapa de su vida se encuentra cree un procedimiento para realizar dicha tarea.

Etapas (E:Edad:Entero, S:Etapas:Cadena)

Edad	Etapas
Entre 0 y 2	Bebé
Entre 3 y 5	Niño
Entre 6 y 12	Pubertad
Entre 13 y 18	Adolescente
Entre 19 y 25	Joven
Entre 26 y 60	Adulto
Mayor a 60	Anciano

Propuesto n.º 74

Enunciado: Dado un número, obtener la suma de sus dígitos pares e impares.

Recuerde: Crear un procedimiento que realice la tarea.

Propuesto n.º 75

Enunciado: Dado un carácter, determinar si es vocal, letra mayúscula, letra minúscula, número o símbolo.

Recuerde: Crear un procedimiento que realice la tarea.

Propuesto n.º 76

Enunciado: Hallar el área de un rectángulo. Cree una función para realizar dicha tarea.

AreaRectangulo(E:Base:Real, E:Altura:Real):Real

Propuesto n.º 77

Enunciado: Un negocio tiene dos tipos de cliente: público en general (G) y cliente afiliado (A). Acepta dos formas de pago: al contado (C) o en plazos (P). Nos piden crear un programa que al ingresar el monto de la compra se obtenga el monto del descuento o el monto del recargo y el total a pagar, según la siguiente tabla:

Tipo	Contado (C) Descuento	Plazos (P) Recargo
Público en general (G)	15 %	10 %
Cliente afiliado (A)	20 %	5 %

- Cree una función para obtener el % de recargo.

Recargo(E:Tipo:Carácter):Real

- Cree una función para obtener el % del descuento.

Descuento(E:Tipo:Carácter):Real

Propuesto n.º 78

Enunciado: Lea un número y devuelva el número en forma inversa. Por ejemplo, si ingresa 123, su número invertido es 321; si ingresa 12345, su número invertido 54321.

Recuerde: Crear una función que realice la tarea.

Propuesto n.º 79

Enunciado: Dada una palabra, determinar si es palíndromo (una palabra es palíndromo si se lee igual de izquierda a derecha o de derecha a izquierda), por ejemplo ANA.

Recuerde: Crear una función que realice la tarea.

Propuesto n.º 80

Enunciado: Cree una matriz de A de 2x2 y otra B de 2x2, y obtenga una matriz $C = A * B$.

Recuerde: Crear una función que realice la tarea.

Impreso en los talleres gráficos de



Surquillo